



```
int iLength, iN;  
double dblTemp;  
bool again = true;  
  
while (again) {  
    iN = -1;  
    again = false;  
    getline(cin, sInput);  
    system("cls");  
    stringstream(sInput) >> dblTemp;  
    iLength = sInput.length();  
    if (iLength < 4) {  
        again = true;  
        continue;  
    } else if (sInput[iLength - 3] !=  
        again = true;  
        continue;  
    } while (++iN < iLength) {  
        if (isdigit(sInput[iN])) {  
            continue;  
        } else if (iN == (iLength - 3) ) {
```



GUÍAS DE PRÁCTICO LENGUAJE DE PROGRAMACIÓN C

Juan Carlos Sandoval Avendaño
2023



GUIAS DE PRÁCTICO LENGUAJE DE PROGRAMACIÓN C.

TEXTO DE APOYO A LA DOCENCIA

Autor:

JUAN SANDOVAL AVENDAÑO

Año: 2023

GUIAS DE PRÁCTICO LENGUAJE DE PROGRAMACIÓN C.

JUAN SANDOVAL AVENDAÑO

Registro de Propiedad Intelectual N° 2023-A-12515 año 2023

ISBN: 987-956-9280-52-8

Prohibida la reproducción total o parcial de esta obra

©UNIVERSIDAD DE CONCEPCION

ÍNDICE

PÁG.

GUÍA DE PRÁCTICO N° 1 HARDWARE Y SOFTWARE	5
GUÍA DE PRÁCTICO N° 2 INTRODUCCIÓN A LA PROGRAMACIÓN	13
GUÍA DE PRÁCTICO N° 3 INTRODUCCIÓN A LA PROGRAMACIÓN EN C	29
GUÍA DE PRÁCTICO N° 4 PROGRAMAS BÁSICOS Y USO DE CÓDIGO ASCII	39
GUÍA DE PRÁCTICO N° 5 TIPOS DE DATOS, VARIABLES, CONSTANTES, OPERADORES, ENTRADA Y SALIDA DE DATOS	59
GUÍA DE PRÁCTICO N° 6 ESTRUCTURAS CONDICIONALES	123

GUÍA DE PRÁCTICO N° 7 CONTADORES, ACUMULADORES DE SUMA Y PRODUCTO, Y ESTRUCTURAS ITERATIVAS	200
GUÍA DE PRÁCTICO N° 8 NÚMEROS SEUDOALEATORIOS	268
GUÍA DE PRÁCTICO N° 9 FUNCIONES	303
GUÍA DE PRÁCTICO N° 10 TIPOS ESTRUCTURADOS	333
GUÍA DE PRÁCTICO N° 11 CADENAS DE CARACTERES	420
GUÍA DE PRÁCTICO N° 12 ARCHIVOS	450
BIBLIOGRAFÍA	474

GUÍA DE PRÁCTICO N° 1

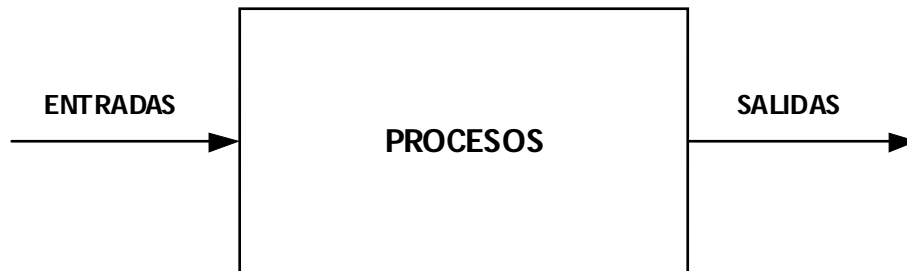
HARWARE Y SOFTWARE

Teoría Hardware

1 Computador.

El hardware es todo lo tangible en un sistema de cómputo.

Un *computador* es una máquina (hardware) que acepta entradas o datos ingresados, los cuales son procesados usando programación (software), y generando salidas, que son los resultados del procesamiento de datos.



▣ 2 Partes de una Computadora

El componente físico más importante de una computadora es la CPU o Unidad Central de Procesos.

Es común que, además de la CPU, un ordenador esté compuesto físicamente por una placa base o tarjeta madre, memoria RAM, disco duro, tarjeta gráfica, teclado y monitor.

En la tarjeta madre se conectan los componentes de la computadora, a saber, memoria RAM, discos duros, tarjeta gráfica o de video, etc.

La memoria RAM o principal (Random Access Memory), es una memoria temporal, porque todo lo que se encuentre en ella desaparece, al desconectar la computadora o si se produce un corte de energía eléctrica, aunque sea muy breve; esto obliga a guardar frecuentemente la información que estamos creando en un medio de almacenamiento permanente o secundario, como es un disco duro (Hard Disk o HD), un disco de estado sólido (Solid State Disk o SSD), un pendrive, un cd, un dvd, etc.

Es común organizar la memoria en grupos de 8 bits denominados bytes. Habitualmente un byte representa un solo carácter (es decir, una letra, un dígito único o un símbolo de puntuación).

La magnitud de la memoria de una computadora se suele expresar como un múltiplo de $2^{10} = 1024$ bytes. Esta cantidad se suele denominar 1 *kB* (kiloByte). La siguiente tabla muestra algunas cantidades comunes al momento de hablar de capacidad de almacenamiento.

Nombre	Equivalencia	Abreviaciones	Equivalencia en bytes
1 Kilobyte	1024 bytes	<i>kB, kbyte</i>	$2^{10} = 1024$ bytes
1 Megabyte	1024 <i>KB</i>	<i>MB</i>	$2^{20} = 1.048.576$ bytes
1 Gigabyte	1024 <i>MB</i>	<i>GB</i>	$2^{30} = 1.073.741.824$ bytes
1 Terabyte	1024 <i>GB</i>	<i>TB</i>	$2^{40} = 1.099.511.627.776$ bytes
1 Petabyte	1024 <i>TB</i>	<i>PB</i>	$2^{50} = 1.1259 \times 10^{15}$ bytes

En la mayoría de los casos, en los discos duros o discos de estado sólido se almacena el sistema operativo y las aplicaciones que utilizará el usuario del computador.

La tarjeta de video permite el procesamiento de información gráfica a visualizar en un monitor. La parte más importante de una tarjeta de video es la GPU (Graphics Processing Unit o Unidad de Procesamiento Gráfico).

3 **Dispositivos Periféricos**

Los *dispositivos periféricos* son los elementos físicos que se conectan a la computadora y pueden ser de entrada (para ingresar información) o de salida (para exhibir la información). La mayoría de los dispositivos periféricos requiere la instalación de un *controlador* o driver para funcionar, que es un programa que permite al sistema operativo interactuar con un periférico, y en la mayoría de los sistemas operativos modernos es instalado de forma automática, sin la intervención directa del usuario.

El dispositivo periférico de entrada común es el teclado y el dispositivo de salida común es el monitor.

Software

4 **Definición de Software.**

El hardware de un computador no es de utilidad si no está acompañado de programas de computador, archivos de datos, etc. A esta colección de información se le llama *software*. En definitiva, el software es todo lo intangible en un sistema informático.

Cuando se guarda información en un disco ésta se almacena en forma de *archivo* o fichero que es una colección de información identificado con una ruta (path) completa y que termina con un nombre único. Este nombre puede contener diversos caracteres, seguido de una *extensión*, la que sirve formalmente para identificar el tipo de archivo.

A continuación se mencionan algunas extensiones y los tipos de archivo asociados.

Extensión (es)	Tipo (s) de archivo
rar, zip, arc, arj, gzip	comprimidos en distintos formatos
gif, jpg, jpeg, tiff, bmp	imagen en distintos formatos
c	fuente en lenguaje C
h	encabezado en lenguaje C
obj	objeto de un compilador
dll	librería de funciones en Windows
hlp	ayuda
txt, doc, docx	texto, documento de Word
xls, xlsx	hoja de cálculo de Excel
ppt, pptx	presentación de PowerPoint
exe, msi	programas ejecutables en Windows
mp3	audio

La información de un archivo puede consistir en un texto (un informe), datos o un programa. El archivo puede tener cualquier tamaño, el cual está limitado sólo por el espacio disponible en el disco en que se está guardando, y por la cantidad de memoria principal (RAM) con que se cuenta.

5 Sistema Operativo

Consideremos ahora, un conjunto de programas que se encargan de administrar los recursos del computador y permitir la comunicación del usuario con el computador, denominado *sistema operativo*. Se necesita un sistema operativo para hacer que la computadora realice acciones.

Existen varios sistemas operativos en distintas plataformas.

Las PC's usan Windows en sus distintas versiones, Linux en sus diferentes distribuciones, etc.

Las Macintosh de Apple emplean las distintas versiones de MacOS

Los smartphones utilizan Android de Google, iOS de Apple, HarmonyOS de Huawei, etc.

EJERCICIOS

HARDWARE Y SOFTWARE

- [1] Defina:
a) GPU
b) CPU
c) Sistema Operativo
d) Hardware
- [2] ¿Cuál es la diferencia entre 400 *MB* y 400 *Mb* (*Mb* : *Megabits*) ?
- [3] ¿Qué es un virus informático ?
- [4] ¿Por qué Windows 10 no solicita, en la mayoría de los casos, un controlador para usar un pendrive?
- [5] ¿Cuántos archivos MP3 puede almacenar en un pendrive de 128 *GB*? Asuma que cada archivo MP3 ocupa 3.75 *MB* de almacenamiento.
- [6] Si la capacidad máxima de un pendrive es 16 *GB*, entonces ¿cuántos caracteres puede almacenar en él?.
- [7] Suponga que ha escrito 7 páginas de la tesis, cuando de repente se corta el suministro de energía eléctrica. ¿Qué sucede si usted no ha guardado la información en disco duro o en un *ssd*, o en algún otro medio de almacenamiento secundario?
¿Qué medidas adoptaría usted para evitar perder información ?

[8] Mencione 3 periféricos de entrada y 3 de salida

[9] Averigüe cuánta memoria RAM posee el equipo que usa habitualmente, y cuánto espacio de almacenamiento secundario.

[10] ¿Qué es Ubuntu ?

[11] ¿Con qué archivos está asociada la extensión .AVI ?

[12] ¿Puedo abrir un archivo .EXE en Word ? ¿Qué sucede si cambio su extensión a .DOC ?

[12] Mencione 5 archivos ejecutables del sistema operativo que usted usa. ¿Cuál es la ruta para encontrar tales archivos?

[13] Usando el Bloc de Notas que debería estar en el grupo Accesorios de Windows, abra un archivo .DLL ¿Qué caracteres aparecen al inicio del archivo? ¿Entiende algo?
Haga lo mismo anterior, pero con archivos .EXE , .JPG , .DOC , .XLSX

[14] ¿Qué es un bug en programación? ¿Qué es un debugger?

Observaciones:

1) Busque en Internet la información que requiera, para responder las preguntas anteriores.

2) En Windows por defecto la extensión de los archivos no aparece. Averigüe la forma de que las extensiones de los archivos sean visibles.

PROBLEMAS RESUELTOS HARDWARE Y SOFTWARE

2

¿Cuál es la diferencia entre 400 MB y 400 Mb (Mb : Megabits)?

Solución:

Tenemos que $8 \text{ bits} = 1 \text{ byte}$, por lo que

$$400 \text{ MB} = 400 \times 1024 \text{ bytes} = 409600 \text{ bytes} = 409600 \times 8 \text{ bits} = 3276800 \text{ bits}$$

$$400 \text{ Mb} = 400 \times 1024 \text{ bits} = 409600 \text{ bits} \quad \square$$

3

¿Qué es un virus informático?

Solución:

Un virus es un programa informático o código malicioso cuyo fin es alterar el normal funcionamiento de un equipo, dañando o modificando habitualmente archivos del sistema operativo .

6

Si la capacidad máxima de un pendrive es 16 GB, entonces ¿cuántos caracteres puede almacenar en él?

Solución:

Tenemos que 1 carácter es igual a 1 *byte*, por lo que

$$16 \text{ GB} = 16 \times 1024 \text{ MB} \times 1024 \text{ kB} \times 1024 \text{ bytes} = 17.179.869.184 \text{ bytes}$$

Lo anterior muestra que, podemos almacenar 17.179.869.184 caracteres en un pendrive de 16 GB

GUÍA DE PRÁCTICO N° 2

INTRODUCCIÓN A LA PROGRAMACIÓN

Teoría Programación de Computadoras

T 6 ¿Qué es Programar una Aplicación Informática?

Programar es crear un conjunto de instrucciones, en un *lenguaje de programación*, para lograr que un dispositivo electrónico realice el objetivo que nos hemos planteado. Si el dispositivo electrónico es una computadora, entonces se habla de *programa informático o aplicación informática*.

T 7 Lenguaje de Programación

Un *lenguaje de programación* es un conjunto de símbolos y reglas, expresadas en instrucciones formales, con las cuales se construye el *código fuente* de una aplicación o programa informático.

T 8 Clasificación de los Lenguajes de Programación

Habitualmente los lenguajes de programación se clasifican en:

a) **Lenguajes de bajo nivel.** En estos las instrucciones ejercen un control directo sobre el hardware (que es lo tangible de nuestro sistema informático, a saber, teclado, monitor, memoria, etc.), por lo que son dependientes de éste. Un ejemplo de lenguaje de bajo nivel es el lenguaje ensamblador o en inglés, assembler.

b) **Lenguajes de alto nivel.** En estos las instrucciones son, en general, independientes del hardware, y además se expresan en un lenguaje más natural para los humanos.

Ejemplos de lenguajes de alto nivel son C, C++, Pascal, Basic, Fortran, Matlab, Java, Python, Visual Basic, Delphi, Javascript, Julia, PHP, Arduino.

Es común usar lenguajes de alto nivel, en desmedro de los de bajo nivel, porque son más fáciles de codificar, y porque, en general sin grandes modificaciones, son portables de un dispositivo a otro.

9 **Compiladores e Intérpretes**

Un *compilador* es una aplicación informática que convierte el código fuente, es decir las instrucciones escritas por un programador usando un lenguaje de programación, en lenguaje de máquina, que es aquel que entiende el computador o el dispositivo que estemos programando.

En este proceso de traducción el compilador revisa el código fuente completo, indagando si obedece a las reglas establecidas por el lenguaje de programación utilizado; en caso de error, este es indicado al programador por el compilador.

Un *intérprete* es un programa informático que analiza el código fuente línea a línea y lo ejecuta directamente.

Un compilador permite la generación de archivos ejecutables (en Windows habitualmente de extensión .EXE), en cambio el intérprete, en general, no lo permite. De lo anterior, se concluye que el programa interpretado necesita del intérprete para ser ejecutado, en cambio el programa compilado no necesita el compilador una vez creado el ejecutable.

Suele ser más cómodo usar un intérprete en lugar de un compilador, aunque los programas compilados se ejecuten, en general, con una rapidez mucho mayor que los programas interpretados.

El compilador, o el intérprete, son, en sí mismos, un programa de computadora que admite un programa de alto nivel (por ejemplo, un programa en C), como dato de entrada y genera un archivo en lenguaje máquina correspondiente como salida. Se tiene que el programa de alto nivel original se denomina el *programa fuente* y el programa en lenguaje máquina resultante, el *programa objeto*. Cada computadora debe tener su propio compilador, o intérprete, para un lenguaje de alto nivel dado. Es el empleo de

los compiladores y de los intérpretes lo que nos permite conseguir la uniformidad y la independencia de la máquina con lenguajes de alto nivel.

T 10 Entornos de Programación

Un *Entorno de Desarrollo Integrado* (EDI) o *Integrated Development Environment* (IDE) en inglés, es una o varias aplicaciones informáticas que permiten al programador editar código fuente, compilar o interpretar este código, enlazar el código fuente con código externo, y depurar el código para eliminar los errores.

Existen muchos entornos de desarrollo disponibles para distintos lenguajes de programación y para distintas plataformas (Windows, Linux, MacOS, Android, iOS). En la tabla siguiente se muestran algunos ejemplos.

Entorno	Lenguaje(s)	Plataforma(s)	Gratis o De Pago
DevC++	C y C++	Windows	Gratis
Delphi	Pascal	Windows	Comercial
Turbo C++	C y C++	Windows	Comercial
CppDroid	C y C++	Android	Gratis
Arduino	Arduino	Multiplataforma	Gratis
Xcode	Swift, Objective-C	MacOs	Gratis
Eclipse	Java y otros	Multiplataforma	Gratis
Matlab	M	Multiplataforma	Comercial
Anjuta	Multilenguaje	Linux	Gratis
Code::Blocks	C, C++ y Fortran	Multiplataforma	Gratis
Visual Studio Code	Multilenguaje	Multiplataforma	Gratis

T 11 Algoritmos. Diagrama de Flujo y Seudolenguaje.

Un *algoritmo* es una secuencia ordenada de pasos elementales, en número finito, que permiten resolver un problema.

Los pasos que seguimos para preparar un café es un algoritmo, o la secuencia de pasos para obtener las raíces de una ecuación cuadrática es también un algoritmo y el conjunto de etapas para calcular el promedio de un estudiante también es un algoritmo.




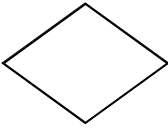
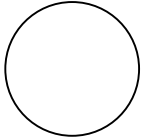
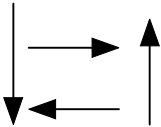
Los algoritmos los podemos expresar mediante el uso de *diagramas de flujo*, que son representaciones gráficas de un algoritmo, o mediante el uso de *seudolenguaje o seudocódigo* que es una representación textual, simbólica, e informal del algoritmo, cercano a nuestro lenguaje natural.

A los diagramas de flujo se les llama de tal manera, porque los símbolos utilizados se conectan por medio de flechas para indicar la secuencia de operación.

Los diagramas de flujo están formados por símbolos gráficos estandarizados, en cambio el seudocódigo no está asociado a una forma estándar de escribir las instrucciones.

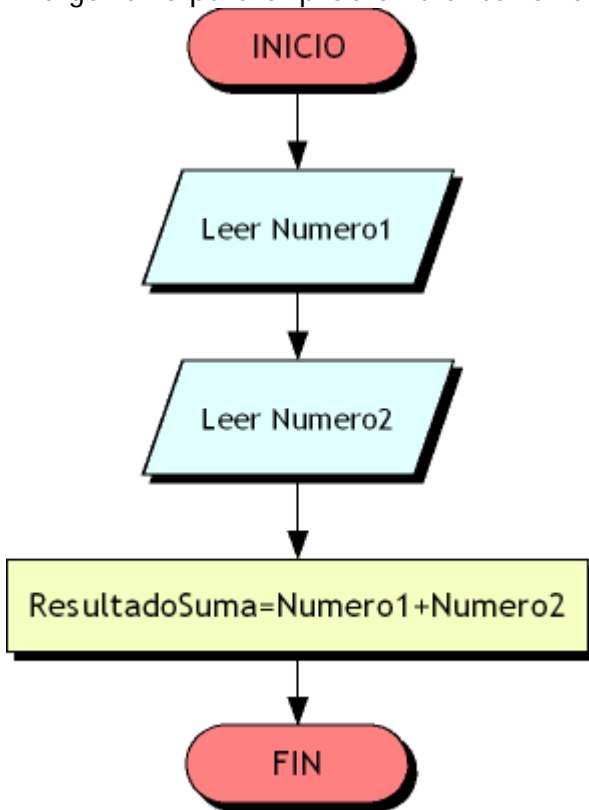
El proceso que permite traducir un algoritmo, expresado como diagrama de flujo o seudolenguaje, a un lenguaje de programación es llamado *codificación*.

Símbolos comunes usados en un diagrama de flujo

Símbolo	Significado	Seudolenguaje
	Inicio, Fin	Inicio, Fin
	Ingreso y salida de datos	Leer x, Escribir x
	Proceso	
	Decisión	Si..., entonces...; sino...
	Conector	
	Flechas que indican el sentido del flujo	

A modo de ejemplo, supongamos que nuestro problema es calcular la suma de dos números.

El algoritmo para el problema anterior expresado como diagrama de flujo sería:



El algoritmo para el problema anterior expresado como pseudocódigo sería, por ejemplo:

- 1) Inicio
- 2) Leer Número1
- 3) Leer Número2
- 4) Calcular: Número1 + Número2
- 5) Fin

EJERCICIOS INTRODUCCIÓN A LA PROGRAMACIÓN

[1] Analice, previa búsqueda en Internet, si son ciertas las siguientes afirmaciones. Justifique las falsas.

- a) No existen intérpretes de C, pero sí de Basic
- b) GCC vale 20 dólares para uso no académico
- c) Octave es un intérprete y Maxima es un compilador
- d) El único compilador de C que existe es GCC
- e) El ANSI C es otro compilador de C
- f) wxDevC++ es un compilador
- g) Ansi C es un EDI
- h) CppDroid es un IDE para el sistema operativo Android
- i) Algunos algoritmos se pueden codificar como diagramas de flujo
- j) El lenguaje de Matlab es interpretado

[2] Mencione dos entornos de programación que sean gratuitos y multiplataforma.

[3] Escriba un algoritmo que calcule el cociente de dos números y que indique cuándo se produce división por cero.

[4] Para el sistema operativo Windows, mencione dos entornos de desarrollo adicionales para programar en lenguaje C.

[5] Para el sistema operativo Android, mencione un entorno de desarrollo adicional para programar en C.

[6] Busque en internet archivos fuente creados con Matlab, Octave, C, C++, Object Pascal, Pascal, Java, Javascript y Python. Revise su contenido con un editor de texto, por ejemplo Notepad. ¿Qué puede concluir, al compararlos?

[7] Escriba el diagrama de flujo correspondiente para resolver los siguientes problemas:

a) Sumar dos números

b) Calcular la suma de los primeros 100 números naturales

c) Determinar el mayor de dos números

d) Calcular el perímetro $P = 2\pi r \operatorname{tg}\left(\frac{\pi}{n}\right)$, de un polígono de n lados circunscrito en una circunferencia de radio r

e) Calcular las raíces de la ecuación cuadrática $ax^2 + bx + c = 0$, usando la fórmula $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

Asuma que a es distinta de cero

f) Calcular la distancia desde el punto conocido (x_0, y_0, z_0) al plano también conocido $ax + by + cz = d$

g) Calcular el cociente A/B de dos números A y B . Si $B = 0$, entonces debemos ingresar el valor de B de nuevo

[8] Escriba en pseudolenguaje lo necesario para:

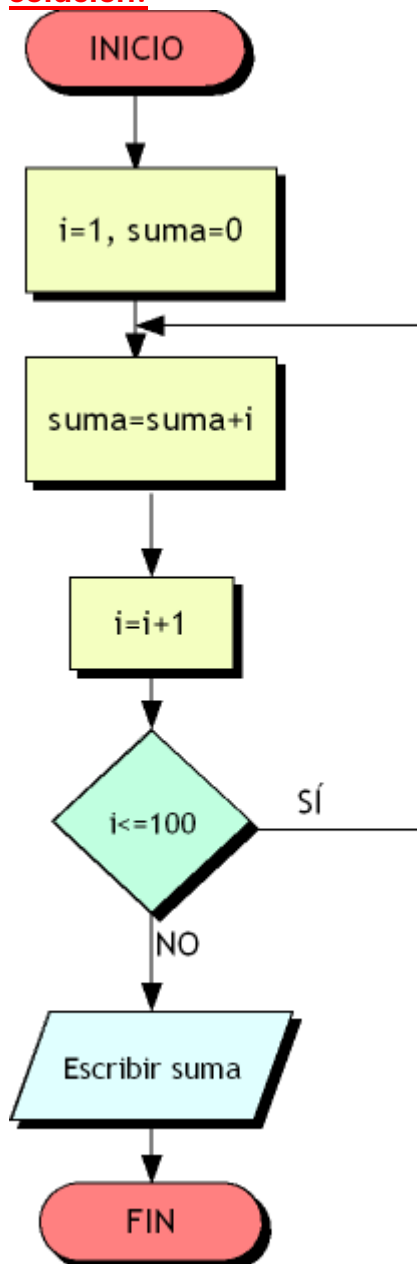
- a) sumar dos números
- b) restar dos números positivos
- c) multiplicar tres números
- d) calcular el área y el perímetro de un círculo de radio 4
- e) calcular el promedio de tres notas
- f) determinar si un número entero es par o impar
- g) calcular el cuadrado y el cubo de un número
- h) convertir una temperatura dada en grados Celsius a grados Fahrenheit ($F=9/5*C + 32$)
- i) convertir una temperatura dada en grados Fahrenheit a grados Celsius
- j) calcular el área de un triángulo cuyos lados tienen longitudes conocidas L_1 , L_2 y L_3
(Use la fórmula : $A = \sqrt{S(S - L_1)(S - L_2)(S - L_3)}$, con $S = \frac{L_1+L_2+L_3}{3}$)
- k) determinar si un número es positivo, negativo ó cero
- l) determinar si el promedio de tres notas es mayor o igual que 4
- m) calcular el cociente A/B de dos números A y B . Si $B = 0$, entonces debemos ingresar el valor de B de nuevo
- n) solicitar al usuario el ingreso de 20 números enteros y mostrar el total de ellos que son pares y la suma de los que son impares

PROBLEMAS RESUELTOS INTRODUCCIÓN A LA PROGRAMACIÓN

7 b)

Escriba el diagrama de flujo para calcular la suma de los primeros 100 números naturales

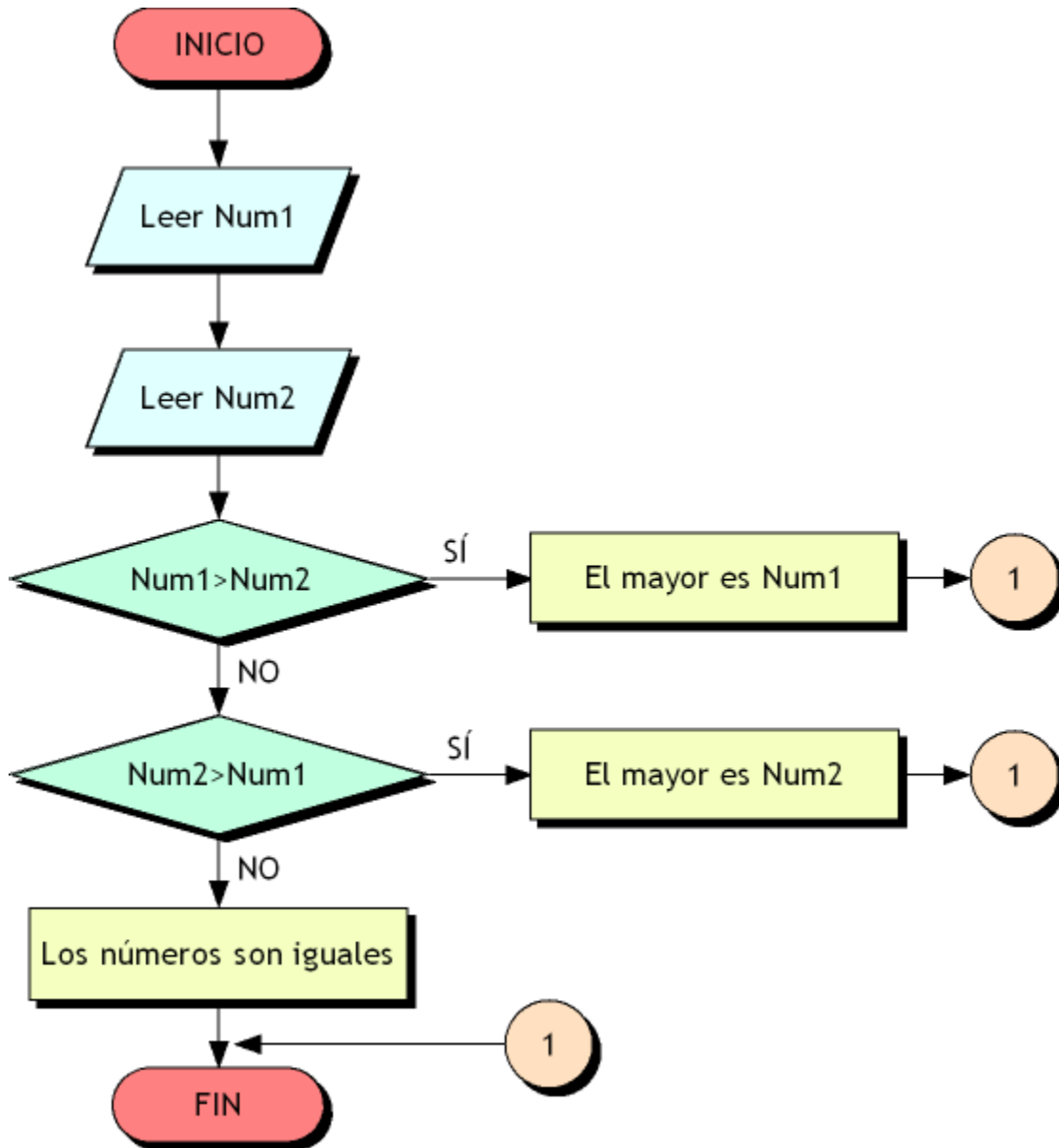
Solución:



7 c)

Escriba el diagrama de flujo para determinar el mayor de dos números

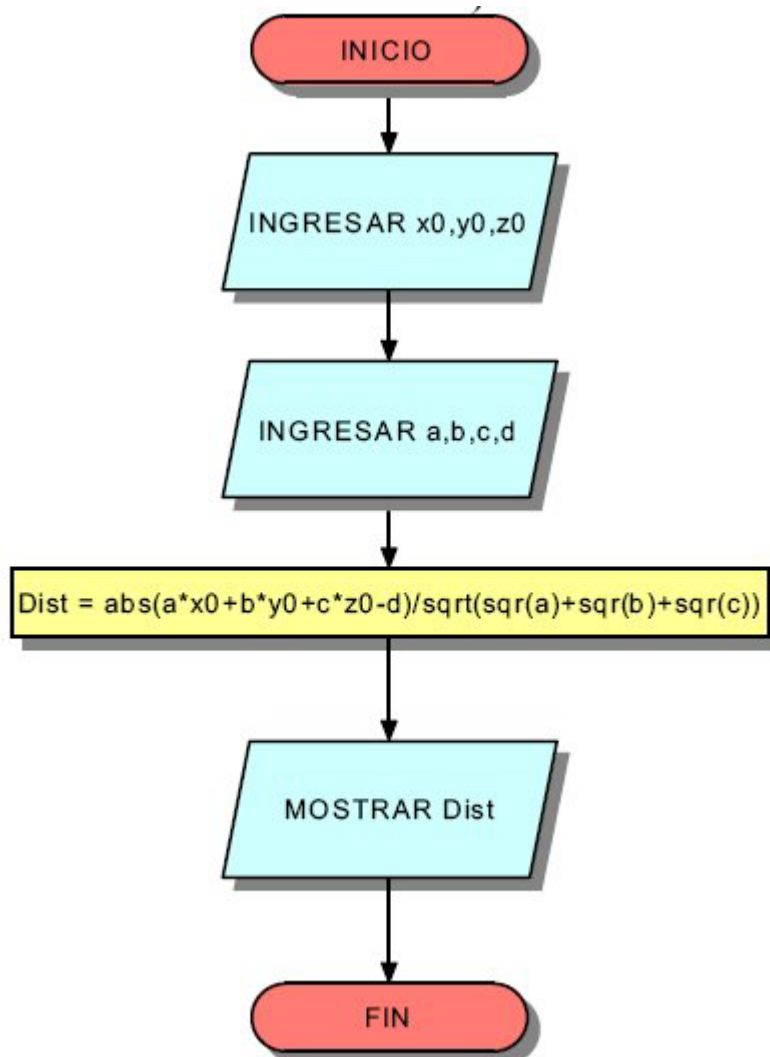
Solución:



7 f)

Escriba el diagrama de flujo para calcular la distancia desde el punto conocido (x_0, y_0, z_0) al plano también conocido $ax + by + cz = d$

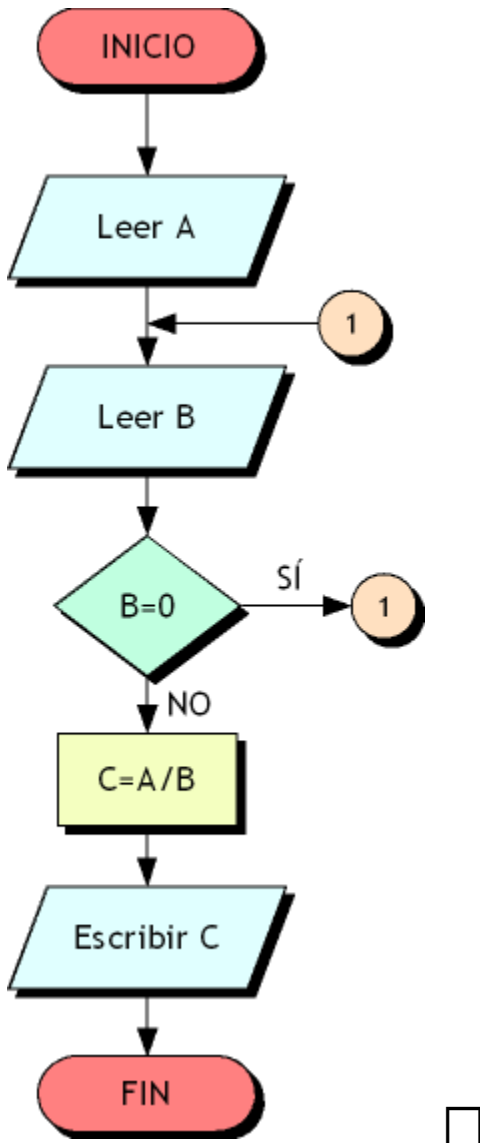
Solución:



7 g)

Escriba el diagrama de flujo para calcular el cociente A/B de dos números A y B . Si $B = 0$, entonces debemos ingresar el valor de B de nuevo

Solución:



8 b)

Escriba en pseudolenguaje lo necesario para restar dos números positivos

Solución:

- 1) Inicio
- 2) Repetir Leer A hasta que A sea positivo
- 3) Repetir Leer B hasta que B sea positivo
- 4) Resta=A-B
- 5) Escribir Resta
- 6) Fin

8 d)

Escriba en pseudolenguaje lo necesario para calcular el área y el perímetro de un círculo de radio 4

Solución:

- 1) Inicio
- 2) $\text{Pi}=3.14159265$
- 3) Radio=4
- 3) $\text{Area}=\text{Pi}*\text{Radio}*\text{Radio}$
- 4) $\text{Perimetro}=2*\text{Pi}*\text{Radio}$
- 5) Escribir Area
- 6) Escribir Perimetro
- 7) Fin

8 e)

Escriba en pseudolenguaje lo necesario para calcular el promedio de tres notas

Solución:

- 1) Inicio
- 2) Leer Nota1, Nota 2 y Nota3
- 3) Promedio= $(\text{Nota1}+\text{Nota2}+\text{Nota3})/3$
- 4) Escribir Promedio
- 5) Fin

8 f)

Escriba en pseudolenguaje lo necesario para determinar si un número entero es par o impar

Solución:

- 1) Inicio
- 2) Leer Numero
- 3) Si Numero dividido por 2 tiene un resto igual a 0, entonces escribir "El número es par", en caso contrario escribir "El número es impar"
- 4) Fin

8 h)

Escriba en pseudolenguaje lo necesario para convertir una temperatura dada en grados Celsius a grados Fahrenheit ($F=9/5*C + 32$)

Solución:

- 1) Inicio
- 2) Leer TemperaturaC
- 3) $TemperaturaF=9/5*TemperaturaC+32$
- 4) Escribir TemperaturaF
- 5) Fin

8 j)

Escriba en pseudolenguaje lo necesario para calcular el área de un triángulo cuyos lados tienen longitudes conocidas L_1, L_2 y L_3

(Use la fórmula : $A = \sqrt{S(S - L_1)(S - L_2)(S - L_3)}$, con $S = \frac{L_1+L_2+L_3}{3}$)

Solución:

- 1) Inicio
- 2) Leer L_1, L_2 y L_3
- 3) $S = (L_1 + L_2 + L_3)/3$
- 3) $A = \sqrt{S*(S - L_1)*(S - L_2)*(S - L_3)}$
- 4) Escribir A
- 5) Fin

8 m)

Escriba en pseudolenguaje lo necesario para calcular el cociente A/B de dos números A y B . Si $B = 0$, entonces debemos ingresar el valor de B de nuevo.

Solución:

- 1) Inicio
- 2) Leer A
- 3) Leer B
- 4) Si B es igual a 0, entonces leer B de nuevo, en caso contrario seguir.
- 5) Calcular el cociente de A y B .
- 6) Escribir el cociente
- 7) Fin

8 n)

Escriba en pseudolenguaje lo necesario para solicitar al usuario el ingreso de 20 números enteros y mostrar el total de ellos que son pares y la suma de los que son impares.

Solución:

- 1) Inicio
- 2) Suma = 0
- 3) Contador = 0
- 4) Para i desde 1 hasta 20 hacer
Leer x
Si x es par entonces Contador = Contador + 1
en caso contrario Suma = Suma + x
- 5) Escribir el total de números pares
- 6) Escribir la suma de los números impares
- 7) Fin

GUÍA DE PRÁCTICO N° 3

INTRODUCCIÓN A LA PROGRAMACIÓN EN C

Teoría

Introducción a la programación en C

T 12 Lenguaje C

C es un lenguaje de programación de *alto nivel*, y de *propósito general*, es decir, con él podremos construir programas de distinto tipo.

Fue creado en 1972 por Ken Thompson y Dennis M. Ritchie en los Laboratorios Bell como evolución del anterior lenguaje B, a su vez basado en BCPL.

C es un lenguaje secuencial, en donde cada sentencia se ejecuta una tras otra, a menos que cambiemos el flujo del programa mediante instrucciones apropiadas para tal efecto.

Usaremos un *compilador* de C llamado GCC (GNU Compiler Collection - Colección de compiladores GNU) que es software libre creado por el proyecto GNU. Este compilador forma parte del Entorno de Desarrollo Integrado (IDE, en inglés) DevC++, que es el entorno en donde trabajaremos en este libro, para crear las aplicaciones escritas en C.

T 13 Entorno de Desarrollo Integrado DevC++

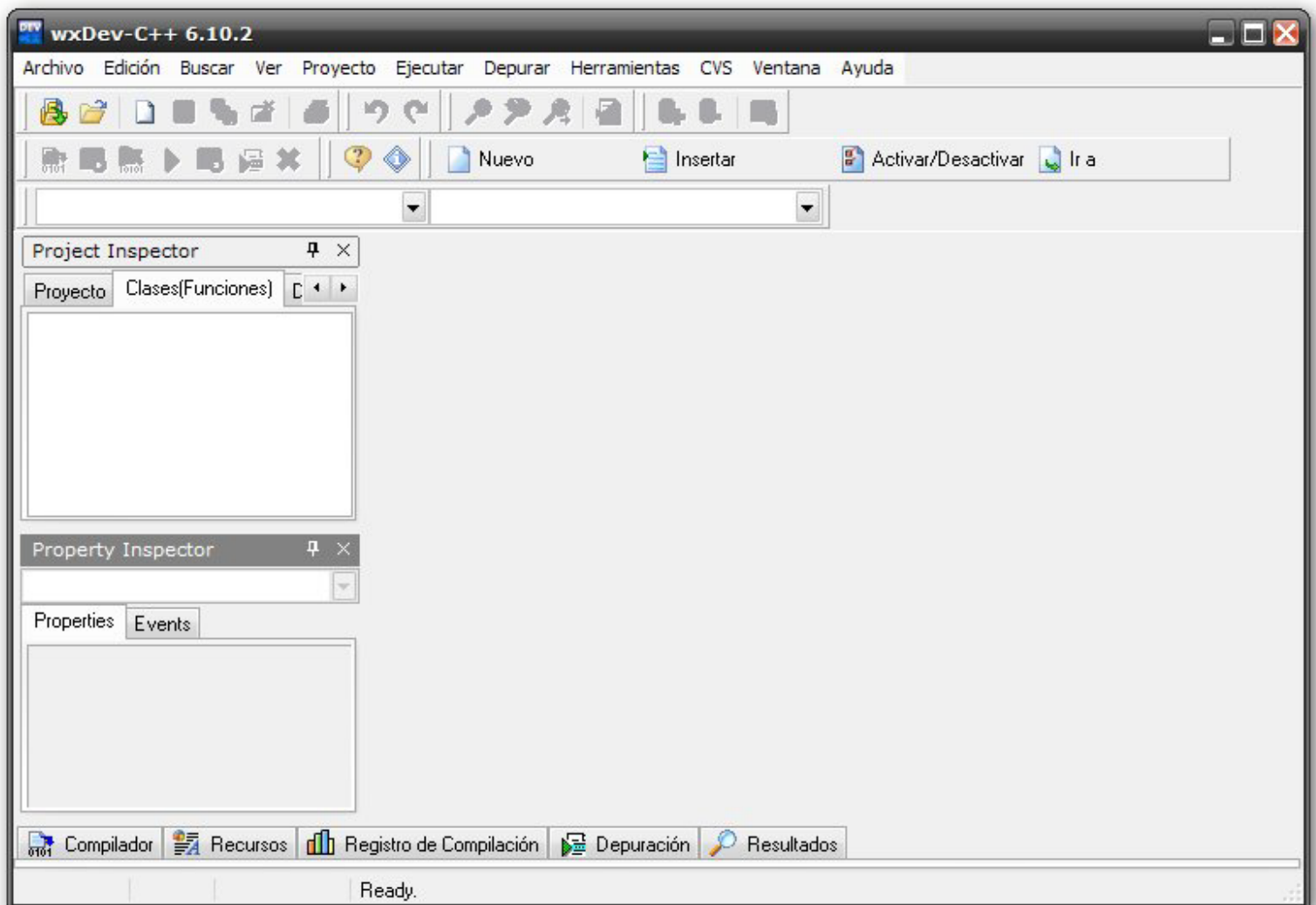
El entorno DevC++ nos permitirá crear un archivo ejecutable (de extensión .EXE) a partir de uno o varios archivos fuente (de extensión .C), que son los archivos de texto que contendrán las distintas sentencias que queremos que se ejecuten para lograr el objetivo deseado.

El DevC++ es un entorno o *ambiente de desarrollo integrado* (Integrated Development Environment IDE) esto es, un conjunto de programas, entre los que se cuentan:

- a) un *editor* que nos permitirá escribir los programas en el lenguaje C
- b) un *compilador* (en inglés, *compiler*) que creará un archivo que la máquina puede interpretar (código objeto) a partir del programa escrito en el editor (código fuente)

- c) un *enlazador* (en inglés, *linker*) que es un programa que toma los archivos de código objeto generado en los primeros pasos del proceso de compilación, la información de todos los recursos necesarios (bibliotecas), quita aquellos recursos que no necesita, y enlaza el código objeto con su(s) biblioteca(s) con lo que finalmente produce un archivo ejecutable o una nueva biblioteca
- d) un *depurador* (en inglés, *debugger*) que es una herramienta para depurar o limpiar un programa de errores.

Después de instalar el DevC++ en un pendrive o en el disco duro, podemos ejecutar el entorno de desarrollo integrado, haciendo doble clic sobre el archivo DEVCPP.EXE o el icono asociado a la aplicación. Al ejecutar tal entorno aparece en la pantalla una distribución más o menos así:

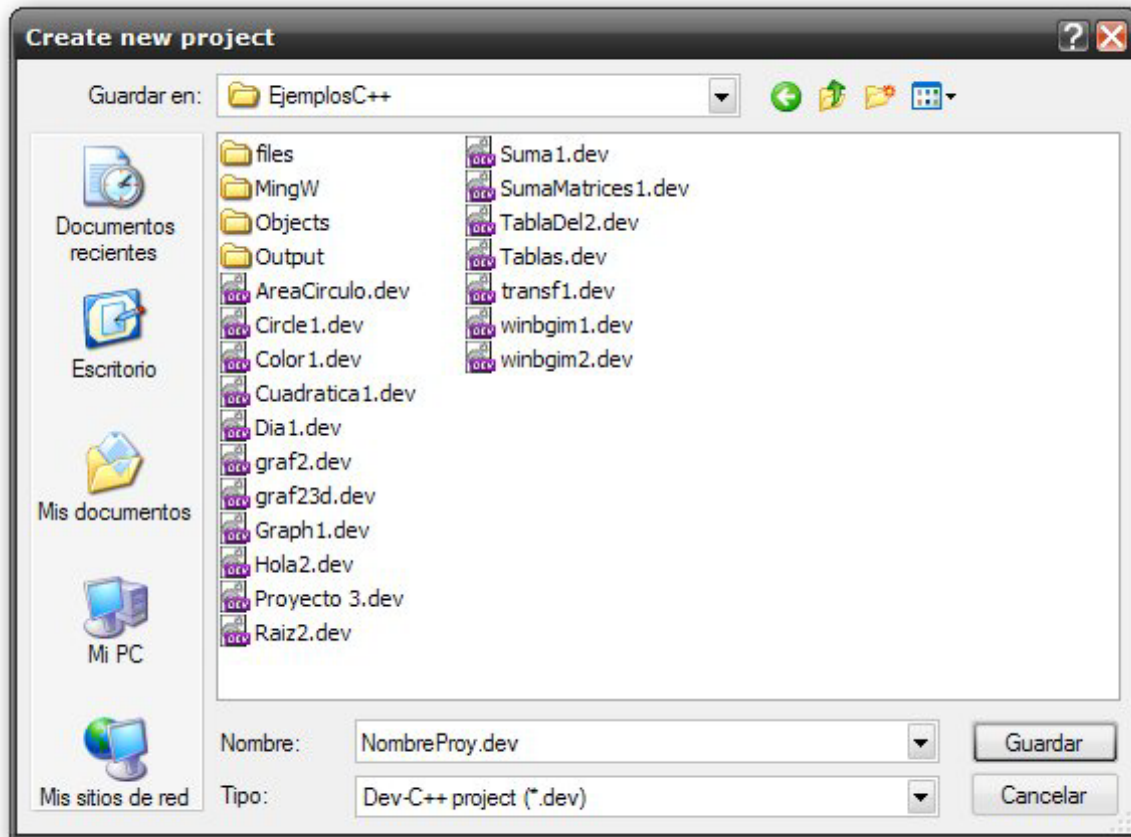


En la parte superior de la pantalla se encuentra el menú principal, que contiene varias opciones, entre las que se cuenta **Archivo** que nos permite acceder a la creación de un nuevo proyecto. Al elegir un nuevo proyecto, después de apuntar Console Application, aparece la ventana



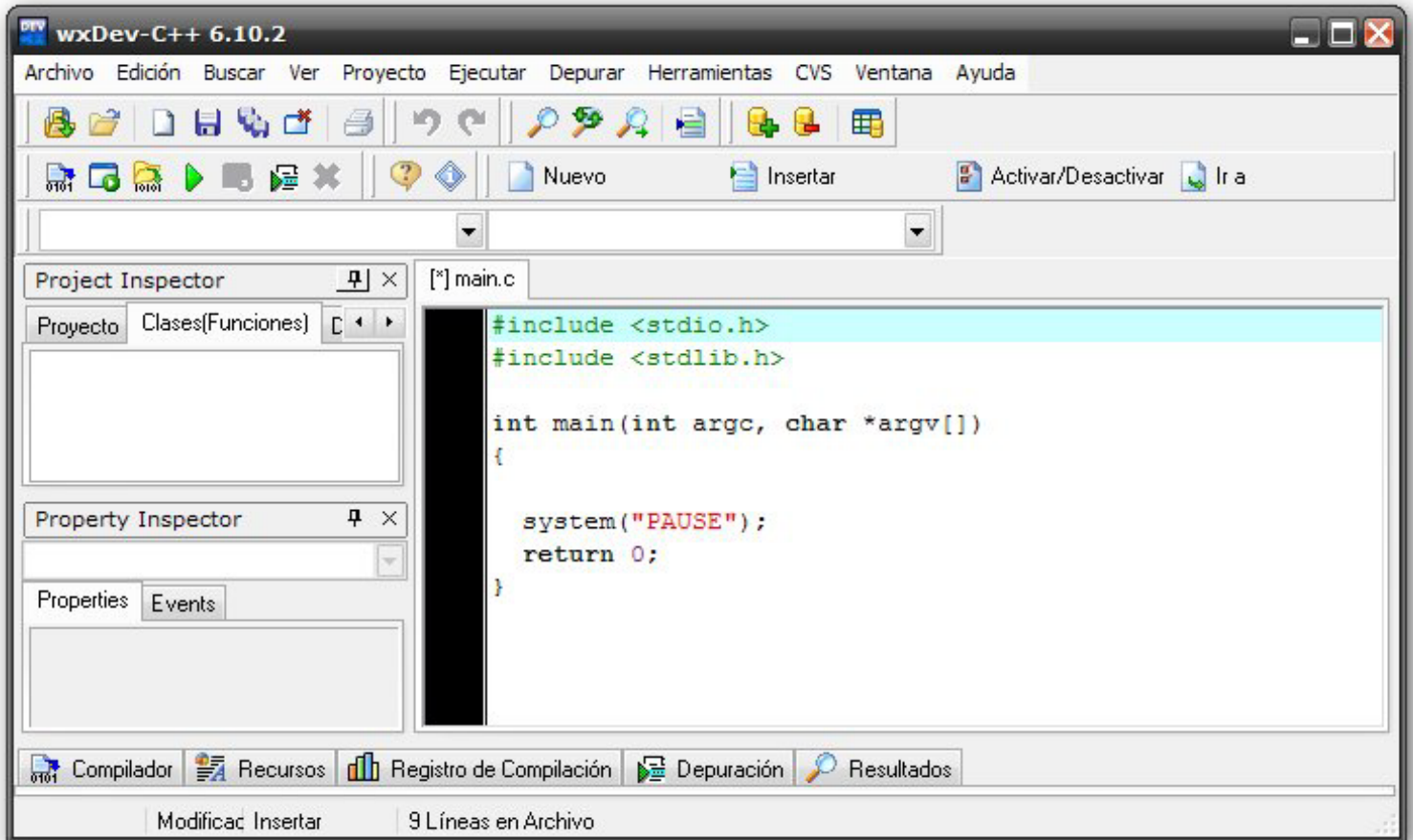
En este libro sólo crearemos aplicaciones de consola (Console Application) escritas en lenguaje C, que será el lenguaje por omisión, usando el entorno de desarrollo DevC++.


Es importante poner un nombre apropiado al proyecto que lo identifique claramente (en el caso de la ventana anterior el nombre asignado es *NombreProy*). Luego de aceptar la creación de un proyecto de consola escrito en C, aparece la ventana



que permite guardar el proyecto en algún directorio elegido por nosotros (si cuentan con un pendrive es recomendable guardarlo en esa unidad).

Luego que se guarda el proyecto, aparece el editor con algunas líneas de código ya escritas que forman parte de la plantilla básica asociada a aplicaciones de consola.



No debemos olvidar lo importante que es guardar la información a disco, para ello, basta con pulsar las teclas Ctrl y S en forma simultánea, o en su defecto elegir las opciones Archivo, Guardar (también sirve hacer clic sobre el icono )

EJERCICIOS

INTRODUCCIÓN A LA PROGRAMACIÓN EN C

[1] a) ¿Cómo se compila un archivo fuente escrito en DevC++?
¿Cómo se ejecuta?

b) ¿Para qué sirve la sentencia
`system("PAUSE");`
que aparece al crear una aplicación de consola usando la plantilla?

c) ¿Es posible usar el editor del DevC++ para crear otros archivos de texto que no sean archivos fuente en C?.

[2] Copie el siguiente archivo fuente en lenguaje C en el editor de DevC++. Notemos que C hace diferencia entre mayúsculas y minúsculas, esto significa que debe transcribir fielmente las líneas siguientes.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define PI 3.141592654

//Programa que calcula el área de un círculo con radio ingresado por el
// usuario

int main(int argc, char *argv[])
{
    float radio, AreaCirculo;
    printf("Ingrese el radio del círculo :");
    scanf("%f",&radio);
    AreaCirculo=PI*pow(radio,2);
    printf("\nEl área del círculo de radio %0.0f es : %f\n",radio,AreaCirculo);
    system("PAUSE");
    return 0;
}
```

Realice todas las acciones que sean necesarias para crear un archivo ejecutable a partir del archivo fuente anterior, identificando claramente la ruta para acceder al ejecutable.

Cierre el entorno de desarrollo y ejecute la aplicación recién creada. ¿Qué puede concluir?

¿Para qué sirve el comando `pow`, y en qué librería está?

[3] Revise la ayuda disponible en el entorno DevC++.

¿Es posible obtener el archivo de ayuda en español? Si su respuesta es afirmativa, obténgalo.

[4] a) Escriba un programa en C que muestre por pantalla el mensaje "El lenguaje C, permite crear aplicaciones"

b) ¿Qué modificaciones le haría al programa para mostrar por pantalla "El lenguaje C," en una línea y "permite crear aplicaciones" en otra?

c) ¿Qué modificaciones le haría al programa para mostrar por pantalla un mensaje distinto?

[5] Para el sistema operativo Windows, mencione dos entornos de desarrollo adicionales gratuitos para programar en lenguaje C. Instale los entornos anteriores y compárelos con DevC++.

[6] Si cuenta con un sistema operativo móvil, instale un entorno de desarrollo gratuito para programar en C, y revíselo.

PROBLEMAS RESUELTOS

INTRODUCCIÓN A LA PROGRAMACIÓN EN C

1

- a) ¿Cómo se compila un archivo fuente escrito en DevC++?
¿Cómo se ejecuta?
- b) ¿Para qué sirve la sentencia `system("PAUSE");` que aparece al crear una aplicación de consola usando la plantilla?
- c) ¿Es posible usar el editor del DevC++ para crear otros archivos de texto que no sean archivos fuente en C?

Solución:

a) Para compilar un archivo fuente escrito en el editor de DevC++ podemos elegir uno de los siguientes procedimientos:

a.1 Ir a la opción *Ejecutar* del menú, luego elegir *Compilar*

a.2 Usando el teclado, pulsar *Ctrl* y *F9* simultáneamente (Esto es lo que se denomina un atajo)

b) La sentencia `system("PAUSE");` sirve para crear una pausa en la ejecución de la aplicación, de modo que el usuario pueda ver la información que muestra tal aplicación. Con esta sentencia aparece por pantalla el mensaje "Presione una tecla para continuar..."

c) Sí es posible crear otros archivos de texto que no sean archivos fuente en C, desde el editor del DevC++; para ello, basta guardarlo con el nombre y extensión que consideremos apropiados

4

- a) Escriba un programa en C que muestre por pantalla el mensaje "El lenguaje C, permite crear aplicaciones"
- b) ¿Qué modificaciones le haría al programa para mostrar por pantalla "El lenguaje C," en una línea y "permite crear aplicaciones" en otra?
- c) ¿Qué modificaciones le haría al programa para mostrar por pantalla un mensaje distinto?

Solución:

a) **Código Fuente**

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    printf("El lenguaje C, permite crear aplicaciones");
    system("PAUSE");
    return 0;
}
```

Salida

```
El lenguaje C, permite crear aplicacionesPresione una tecla para continuar . . . 
```

b) La modificación es la inclusión del carácter de control \n que nos permite avanzar a la siguiente línea.

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    printf("El lenguaje C,\npermite crear aplicaciones");
    system("PAUSE");
    return 0;
}
```

Salida

```
El lenguaje C,
permite crear aplicacionesPresione una tecla para continuar . . .
```

c) La única modificación es cambiar el mensaje por otro.

5

Para el sistema operativo Windows, mencione dos entornos de desarrollo adicionales gratuitos para programar en lenguaje C. Instale los entornos anteriores y compárelos con DevC++.

Solución:

- 1) Visual Studio Code
- 2) Falcon C++

6

Si cuenta con un sistema operativo móvil, instale un entorno de desarrollo gratuito para programar en C, y revíselo.

Solución:

Por ejemplo, para Android CxxDroid y para iOS Mobile C

GUÍA DE PRÁCTICO N° 4

PROGRAMAS BÁSICOS Y USO DEL CÓDIGO ASCII

Teoría

Programas Básicos y Uso del Código ASCII

14 Plantilla

La mayoría de los entornos de programación permiten el uso de plantillas, que incluyen los elementos básicos del lenguaje de programación, necesarios para la construcción de una aplicación. En particular, al crear un programa de consola, DevC++ nos ofrece la siguiente plantilla:

```
#include <stdio.h>
#include <stdlib.h>

int main()
{

    system("PAUSE");
    return 0;
}
```

En la plantilla anterior se han eliminado los argumentos de la función `main()`, porque no haremos uso de ellos en este libro.

Las dos primeras líneas de la plantilla incluyen a las bibliotecas `stdio.h` y `stdlib.h`, de modo que podamos hacer uso de las funciones que componen tales bibliotecas. En particular, la biblioteca `stdlib` incluye la función `system`.

Luego de la inclusión de las bibliotecas aparece la función principal `main()`, que es el punto de partida para la ejecución de nuestro programa, es decir, la

primera sentencia que se ejecutará será la que aparezca inmediatamente después de la llave de apertura en la función `main()`.

Todo programa escrito en lenguaje C debe incluir, a lo menos, una función principal llamada justamente `main()` que devuelve un valor entero igual a 0 si la ejecución ha sido exitosa (`return 0`).

La función `main()` comienza cuando aparece la primera llave (`{`) y termina cuando aparece la llave de cierre final (`}`); además debemos escribir nuestro código justamente entre estas llaves.

15 Salida por Pantalla

La orden `printf` es una función de C, incluida en la biblioteca `stdio.h` que permite, entre otras cosas, mostrar por pantalla usando un formato, lo que se incluye como argumento.

El comando `printf` permite el uso de diferentes *especificadores de formato*, que establecen el formato de salida de los argumentos de la función `printf`. Algunos de los especificadores de formato se muestran en la siguiente tabla

Especificador de formato	Tipo de dato asociado
<code>%f</code>	float (Real)
<code>%e</code> ó <code>%E</code>	float (Real en notación científica)
<code>%i</code>	int (Entero)
<code>%d</code>	int (Entero)
<code>%c</code>	char (un carácter)
<code>%s</code>	string (cadena de caracteres)

El comando `printf` también ofrece la incorporación de secuencias de escape, que permiten mostrar caracteres especiales o realizar ciertas acciones. Algunas de las secuencias de escape se muestran en la siguiente tabla:

Secuencia de escape	Descripción
<code>\n</code>	mueve el cursor al principio de la siguiente línea
<code>\t</code>	mueve el cursor una tabulación horizontal (8 espacios)
<code>\a</code>	alerta (sonido breve)
<code>\b</code>	retrocede un espacio
<code>\'</code>	comilla simple
<code>\"</code>	comilla doble
<code>\\</code>	barra diagonal invertida (backslash)
<code>\?</code>	signo de interrogación

T 16 **Código ASCII**

El código ASCII es un conjunto de equivalencias entre números, caracteres alfanuméricos y otros. Los números van desde 0 hasta 127 en el código ASCII básico, y el extendido se amplía desde 128 a 255.

Para incluir caracteres del código ASCII, es necesario usar el especificador de formato %c que permite reemplazar un caracter por el indicado, mediante el número correspondiente del código, al final del printf.

No es necesario escribir todos los caracteres usando el código ASCII, sólo se utiliza cuando no existe el caracter en el teclado inglés.

Los caracteres del código ASCII extendido de uso común, y que no están disponibles en el teclado inglés de forma directa, son principalmente:

Tildes	Código ASCII
á	160
é	130
í	161
ó	162
ú	163

Eñes	Código ASCII
ñ	164
Ñ	165

Signos de Interrogación y Exclamación	Código ASCII
?	168
!	173

Potencias	Código ASCII
1	251
3	252
2	253

Líneas Simples	Código ASCII
	179
┆	180
┆	191
┆	192
┆	193
┆	194
┆	195
—	196
┆	197
┆	217
┆	218

Líneas Dobles	Código ASCII
	185
	186
	187
	188
	200
	201
	202
	203
	204
=	205
	206

T 17 Color

Para mostrar por pantalla texto en color debemos hacer uso de la función `textcolor(Color)` que está presente en la biblioteca `conio2.h`, que debe ser primero instalada antes de hacer uso de ella.

La función `textcolor(Color)` posee como argumento un color que puede ser cualquiera de los siguientes:

Valor	Argumento	Color
0	BLACK	Negro
1	BLUE	Azul
2	GREEN	Verde
3	CYAN	Cian
4	RED	Rojo
5	MAGENTA	Fucsia
6	BROWN	Café
7	LIGHTGRAY	Gris Claro
8	DARKGRAY	Gris Oscuro
9	LIGHTBLUE	Azul Claro o Brillante
10	LIGHTGREEN	Verde Claro o Brillante
11	LIGHTCYAN	Cian Claro o Brillante
12	LIGHTRED	Rojo Claro o Brillante
13	LIGHTMAGENTA	Fucsia Claro o Brillante
14	YELLOW	Amarillo
15	WHITE	Blanco

Los colores anteriores son datos de tipo entero, es decir, al color negro le corresponde el número 0, al azul el 1, y así sucesivamente, tal como aparece en la tabla anterior.

También podemos utilizar la función `textbackground(Color)`, de la misma biblioteca, que nos permite cambiar el color de fondo del texto, usando los mismos colores de la tabla anterior.

Otra forma de incluir color en una aplicación es hacer uso de la orden `system`, presente en la biblioteca estándar `stdio.h`

La forma de utilizar la orden es

```
system("color FondoTexto")
```

donde Fondo es un dígito hexadecimal que indica el color de fondo y Texto es un dígito hexadecimal que indica el color del Texto, que aparecen en la consola. Los dígitos pueden ser cualquiera de los siguientes valores:

Dígito Hex	Color en Inglés	Color en Español
0	Black	Negro
1	Blue	Azul
2	Green	Verde
3	Aqua	Aguamarina
4	Red	Rojo
5	Purple	Púrpura
6	Yellow	Amarillo
7	White	Blanco
8	Gray	Gris
9	Light Blue	Azul Claro
A	Light Green	Verde Claro
B	Light Aqua	Aguamarina Claro
C	Light Red	Rojo Claro
D	Light Purple	Púrpura Claro
E	Light Yellow	Amarillo Claro
F	Bright White	Blanco Brillante

Por ejemplo, si queremos el fondo blanco con texto verde debemos escribir

```
system("color 72");
```

Los números 7 y 2 deben aparecer sin espacio entre ellos.

T 18 Comentarios

El lenguaje C permite adicionar comentarios en el código fuente, principalmente con el objetivo de documentar el código, es decir informar acerca de lo que hace cada sección de código, con miras a su mantenimiento o reutilización.

Los comentarios no son compilados, es decir, son ignorados por el compilador y por ende, no aparecerán al momento de la ejecución de la aplicación.

Existen dos formas de adicionar comentarios en un programa en C:

- a) // comentario en línea
- b) /* Comentario en bloque */

Como se observa los comentarios en línea comienzan con doble barra diagonal y pueden ser incluidos, por ejemplo, al lado de una sentencia; los comentarios en bloque deben comenzar con `/*` y terminar con `*/`, y lo que esté entre ambos pares de caracteres será el comentario.

A veces los comentarios, especialmente el comentario en línea, son usados para inhabilitar una sección de código a modo de prueba.

EJERCICIOS PROGRAMAS BÁSICOS Y USO DEL CÓDIGO ASCII

- [1]** a) Escriba un programa en C que muestre por pantalla en mensaje "Hola Mundo", sin incluir las comillas dobles.
b) Modifique el programa anterior de modo que aparezcan las comillas dobles.

- [2]** Escriba un programa en C que, usando el código ASCII extendido, muestre por pantalla el siguiente mensaje:

Habrá test de entrada en cada práctico de Lenguaje de Programación.
¡Éxito este año!

- [3]** Escriba un programa en C que muestre por pantalla en color azul claro el siguiente mensaje:

Debo ejercitar para aprobar esta asignatura.!!!!

- [4]** Escriba un programa en C que, usando el código ASCII extendido, muestre por pantalla en color verde el siguiente mensaje:

Mañana habrá prueba de química y matemática.

- [5]** Escriba un programa en C que, usando el código ASCII extendido, muestre por pantalla en color azul brillante el siguiente mensaje:

El entorno de programación a usar en este curso es DevC++ y el compilador asociado es gcc. ¿Habrá otros entornos donde programar en lenguaje C?

[6] Escriba un programa en C que muestre por pantalla las siguientes figuras, usando un color diferente para cada una de ellas :

a)

```
 *
 ***
 *****
 *********
 *****
 ***
 *
```

b)

```
 *
 **
 ***
 ****
 *****
 ****
 ***
 **
 *
```

[7] Escriba un programa en C que muestre por pantalla, usando diferente color para cada línea, lo siguiente:

```
1234567890123456789012345
 *                *
*****          _____
RaReRiRoRu
```

Obs. : El recuadro indica borde de la pantalla. La esquina superior izquierda del recuadro indica esquina superior izquierda de la pantalla luego de ser limpiada.

[8] Escriba un programa en C que muestre, por pantalla, el siguiente mensaje tal cual aparece:

¡¡¡Debo ejercitar todas las semanas para aprobar esta asignatura.!!!!

Es fácil programar con el compilador GCC.

¿Será necesario estudiar todos los días?. ¿Será que el área de un círculo es $\text{Pi} \cdot r^2$?

123456789012345678901234567890123456789012345678901234567890123456789012

Observaciones: Cada una de las líneas de texto debe aparecer con un color distinto. Existen dos tabulaciones en la segunda línea de texto.

[9] Escriba un programa en C que muestre por pantalla, usando diferente color para cada línea de texto, lo siguiente:

```
La mañana está clara.  
¿Cuál es la razón de tanto  
alboroto?  
El área de un círculo es 12  
cm2
```

Incluya el trazado de las líneas que encierran el texto.

[10] Escriba un programa en C que muestre por pantalla lo siguiente:

```
JJJJ      CCCCC  
  JJ      CC  
  JJ      CC  
  JJ      CC  
JJJ       CCCCC
```

PROBLEMAS RESUELTOS

PROGRAMAS BÁSICOS Y USO DEL CÓDIGO ASCII

1

- a) Escriba un programa en C que muestre por pantalla en mensaje "Hola Mundo", sin incluir las comillas dobles.
b) Modifique el programa anterior de modo que aparezcan las comillas dobles.

Solución:

a) **Código Fuente**

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main()
{
    printf("Hola Mundo\n");
    system("PAUSE");
    return 0;
}
```

Salida

```
Hola Mundo
Presione una tecla para continuar . . .
```

Observación:

En este programa, se usa la orden printf para mostrar el mensaje *Hola Mundo*, y se salta a la siguiente línea gracias a la aparición de la secuencia de escape \n



b) **Código Fuente**

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main()
{
```

```
printf("\n\"Hola Mundo\"\n");
system("PAUSE");
return 0;
}
```

Salida

```
"Hola Mundo"
Presione una tecla para continuar . . .
```

Observación:

En este programa, se usa la secuencia de escape \" para mostrar las comillas dobles.

2

Escriba un programa en C que, usando el código ASCII extendido, muestre por pantalla el siguiente mensaje:

Habrá test de entrada en cada práctico de Lenguaje de Programación.
¡Éxito este año!

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main()
{
    printf("\n");
    printf(" Habr%c test de entrada en cada pr%ctico de Lenguaje de Programaci%cn",160,160,162);
    printf("\n\t%c%cxito este a%co!",173,144,164); printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida

```
Habr  test de entrada en cada pr ctico de Lenguaje de Programaci n.  
 xito este a o!
```

```
Presione una tecla para continuar . . .
```

Observaciones:

1) La sentencia `printf("\n");` permite avanzar una l nea antes de escribir el mensaje solicitado.

2) En este programa, se usan tantos especificadores de formato `%c` como caracteres del c digo ASCII extendido queremos agregar.

Por ejemplo, en la l nea de c digo

```
printf("Habr%c test de entrada en cada pr%cctico de Lenguaje de Programaci%c",160,160,162);
```

el primer `%c` se corresponde con el c digo 160 que es la letra a con tilde, el segundo con la siguiente a con tilde, y el  ltimo con la o con tilde que tiene c digo num rico 162 (Ver [T](#) 16)

3) La letra E con tilde ( ) tiene c digo ascii igual a 144 [T](#)

3

Escriba un programa en C que muestre por pantalla en color azul claro el siguiente mensaje:

Debo ejercitar para aprobar esta asignatura.!!!!

Soluci n:

C digo Fuente

```
#include <stdio.h>  
#include <stdlib.h>  
#include <conio2.h>  
int main()
```

```
{
    textcolor(LIGHTBLUE);
    textbackground(WHITE);
    printf("\nDebo ejercitar para aprobar esta asignatura.!!!!\n\n");
    system("PAUSE");
    return 0;
}
```

Salida

```
Debo ejercitar para aprobar esta asignatura.!!!!
Presione una tecla para continuar . . .
```

Observaciones:

1) En este programa se usa la sentencia `textcolor(LIGHTBLUE);` para mostrar, en color azul claro, el texto que sigue a tal sentencia; además se usa la sentencia `textbackground(WHITE);` porque en este libro el fondo elegido para la salida es el blanco, y no el negro que es el color de fondo por defecto. La función `textcolor` pertenece a la biblioteca `conio2` que ha sido incluida al inicio del programa.

2) La sentencia `textcolor(LIGHTBLUE);` cambia el color a azul claro para todo lo que se muestre a continuación de tal sentencia; es por esto que el texto "Presione una tecla para continuar ..." aparece también en color azul claro.

3) Para instalar `conio2.h`, primero debemos descargar la biblioteca desde internet, por ejemplo desde github (recomiendo el enlace <https://github.com/Fernando-Lafeta/Biblioteca-Conio-2> actualmente disponible para descarga)

Luego copiar el archivo `conio2.h` a la carpeta `C:\Dev-Cpp\include`. Enseguida copiar el archivo `libconio.a` a la carpeta `C:\Dev-Cpp\lib`

Finalmente, para lograr que la biblioteca `conio2.h` sea enlazada al ejecutable debemos ir a la opción del menú Herramientas, luego elegir Opciones del Compilador y agregar lo que se muestra en la siguiente figura



Luego de hacer clic sobre el botón Aceptar, hemos añadido el comando `-lconio` al enlazador (linker), de modo que la biblioteca `conio2` sea adicionada al ejecutable.

6

Escriba un programa en C que muestre por pantalla las siguientes figuras, usando un color diferente para cada una de ellas :

a)

```
*
***
*****
*****
*****
***
*
```

b)

```
*
**
***
****
*****
*****
*****
*****
*****
```

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <conio2.h>
```

```
int main()
{
    textbackground(WHITE);
    textcolor(GREEN);
    printf(" *\n");
    printf(" **\n");
    printf(" ***\n");
    printf(" ****\n");
    printf("*****\n");
    printf(" *****\n");
    printf("  ***\n");
    printf(" *\n");
    printf("\n\n");
    textcolor(BLUE);
    printf("*\n");
    printf("**\n");
    printf("***\n");
    printf("****\n");
    printf("*****\n");
    printf("*****\n");
    printf("*****\n");
    printf("*****\n");
    printf("\n");
    system("PAUSE");
```



```
return 0;  
}
```

Salida

```
      *  
     ***  
    *****  
 *****  
    *****  
     ***  
      *  
  
*  
**  
***  
****  
*****  
*****  
*****  
*****  
*****  
*****  
  
Presione una tecla para continuar . . .
```

Observaciones:

- 1) Las figuras solicitadas se lograron insertando tantos espacios como fueran necesarios usando la tecla espaciadora.
- 2) La orden `textbackground(WHITE)`; permite cambiar el color de fondo del texto mencionado.
- 3) Se incluyó la línea de código `printf("\n\n");` para separar la primera figura de la segunda.

8

Escriba un programa en C que muestre, por pantalla, el siguiente mensaje tal cual aparece:

!!!Debo ejercitar todas las semanas para aprobar esta asignatura!!!!

Es fácil programar con el compilador GCC.

¿Será necesario estudiar todos los días?. ¿Será que el área de un círculo es $\text{Pi} \cdot r^2$?

1234567890123456789012345678901234567890123456789012345678901234567890123456789012

Observaciones: Cada una de las líneas de texto debe aparecer con un color distinto. Existen dos tabulaciones en la segunda línea de texto.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <conio2.h>

int main()
{
    printf("\n");
    textbackground(WHITE);
    textcolor(LIGHTRED);
    printf("%c%c%cDebo ejercitar todas las semanas para aprobar esta asignatura.!!!",173,173,173);
    textcolor(LIGHTCYAN);
    printf("\n\n\t\tEs fácil programar con el compilador GCC",160);
    textcolor(LIGHTBLUE);
    printf("\n\n%cSer%c necesario estudiar todos los días?.",168,160,161);
    printf(" %cSer%c que el área de un círculo es Pi*r^2?",168,160,160,161,253);
    textcolor(LIGHTMAGENTA);
    printf("\n\n1234567890123456789012345678901234567890123456789012345678901234567890123456789012");
    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida

```
!!!Debo ejercitar todas las semanas para aprobar esta asignatura.!!!
      Es fácil programar con el compilador GCC
¿Será necesario estudiar todos los días?. ¿Será que el área de un círculo es Pi*r^2?
1234567890123456789012345678901234567890123456789012345678901234567890123456789012
Presione una tecla para continuar . . .
```

Observaciones:

1) El signo de apertura de exclamación ¡ se debe incluir usando el código ASCII 173, en cambio el signo de cierre de exclamación se puede escribir directamente, pues aparece en el teclado en inglés.

- 2) En donde aparezca, la secuencia de escape `\t` permite incluir una tabulación, es decir, 8 espacios.
- 3) El cuadrado de `r` puede ser insertado usando el código ASCII extendido 253.
- 4) Es posible escribir una sola línea de texto usando tantos `printf` como se quiera. Por ejemplo, para la tercera línea de texto se usaron dos `printf`.

9

Escriba un programa en C que muestre por pantalla, usando diferente color para cada línea de texto, lo siguiente:

```

La mañana está clara.
¿Cuál es la razón de tanto
alboroto?
El área de un círculo es 12
cm2
```

Incluya el trazado de las líneas que encierran el texto.

Solución:

Código Fuente

```

#include <stdio.h>
#include <stdlib.h>
#include <conio2.h>

int main()
{
    textbackground(WHITE);
    //Línea superior que encierra el texto
    printf("%c%c%c%c%c%c%c%c%c%c",218,196,196,196,196,196,196,196,196,196);
    printf("%c%c%c%c%c%c%c%c%c%c",196,196,196,196,196,196,196,196,196,196);
    printf("%c%c%c%c%c%c%c%c%c%c",196,196,196,196,196,196,196,196,191);
    // Primera línea de texto, incluyendo los bordes laterales
    textcolor(LIGHTRED);
    printf("\n%cLa ma%cana est%c clara.%7c",179,164,160,179);
    /* Segunda línea de texto, incluyendo los bordes laterales */
    textcolor(LIGHTCYAN);
    printf("\n%c%cCu%cl es la raz%cn de tanto %c",179,168,160,162,179);
    printf("\n%calboroto?%19c",179,179);
    /* Tercera línea de texto, incluyendo los bordes laterales */
    textcolor(GREEN);
    printf("\n%cEl %carea de un c%rculo es 12%c",179,160,161,179);
    printf("\n%ccm%c%25c",179,253,179);
}
```

```

//Linea inferior que encierra el texto
printf("\n%c%c%c%c%c%c%c%c%c%c",192,196,196,196,196,196,196,196,196,196);
printf("%c%c%c%c%c%c%c%c%c%c",196,196,196,196,196,196,196,196,196,196);
printf("%c%c%c%c%c%c%c%c%c",196,196,196,196,196,196,196,217);
textcolor(LIGHTGRAY);
printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida

```

La mañana está clara.
¿Cuál es la razón de tanto
alboroto?
El área de un círculo es 12
cm²

```

Presione una tecla para continuar . . .

Observaciones:

- 1) Para dibujar cajas con bordes simples se usa el código ASCII extendido para líneas simples (Ver [T 16](#))
- 2) Note que LIGHTGRAY es el color por defecto para las salidas, y es el que se utiliza en este programa para que aparezca el mensaje "*Presione una tecla para continuar ...*" que se obtiene gracias a la sentencia `system("PAUSE");`
- 3) El formato `%7c`, por ejemplo, de la línea que muestra el mensaje "La mañana está clara.", permite avanzar 6 espacios para luego mostrar el carácter 179 del código ASCII que es una línea vertical.
- 4) En este programa se han incluido comentarios, para describir la funcionalidad de cada sección de código.

GUÍA DE PRÁCTICO N° 5

TIPOS DE DATOS, VARIABLES, CONSTANTES, OPERADORES Y, ENTRADA Y SALIDA DE DATOS

Teoría

Tipos de Datos, Variables, Constantes, Operadores y, Entrada y Salida de Datos

T 19 Tipos de Datos

El tipo de dato nos entrega información sobre la clase de dato que se va a procesar en el programa, por ejemplo, existen datos de tipo real y otros de tipo entero.

Todo lenguaje de programación permite manejar distintos tipos de datos; en particular, el lenguaje C cuenta con los siguientes tipos de datos básicos:

Tipo de Dato	Se escribe en C como	Especificador de Formato
Entero	int	%i ó %d
Real	float	%f
Real de doble precisión	double	%f
Un caracter	char	%c
Cadena de caracteres	char[n]	%s

A continuación, se muestran los tipos de datos básicos, junto con la cantidad de memoria que ocupan y el rango que abarcan. La cantidad de memoria ocupada y el rango pueden variar si cambiamos de compilador.

Tipo de Dato	Memoria	Rango
int	32 bits	[- 2.147.483.648, 2.147.483.647]
float	32 bits	[3.4×10^{-38} , 3.4×10^{38}] (7 a 8 decimales)
double	64 bits	[1.7×10^{-308} , 3.4×10^{308}] (15 a 16 decimales)
char	8 bits	[- 128, 127]

Con algunos tipos de datos podemos usar modificadores, que permiten ajustar ciertas propiedades de cada tipo. A continuación, se muestran algunos tipos de datos con sus modificadores.

Tipo de Dato	Memoria	Rango
short int	16 bits	[- 32.768, 32.767]
unsigned int	32 bits	[0, 4.294.967.295]
long double	80 bits	[3.4×10^{-4932} , 1.1×10^{4932}] (20 decimales aprox.)
unsigned char	8 bits	[0, 255]

T 20 **Variables y Constantes**

Una variable es una entidad que permite almacenar un valor que puede cambiar durante la ejecución de un programa, a diferencia de una constante cuyo valor será el mismo a lo largo de todo el programa.

Antes de utilizar una variable o una constante ésta debe ser declarada, indicando el tipo de dato asociado y el nombre.

El nombre se conoce como identificador y debe obedecer ciertas reglas:

- a) Sólo se pueden usar letras (mayúsculas o minúsculas), números y ciertos caracteres no alfanuméricos, como el '_', pero nunca un punto, coma, guión, comillas o símbolos matemáticos o interrogaciones.
- b) El primer caracter no puede ser un número.
- c) C distingue entre mayúsculas y minúsculas, de modo que los identificadores par y Par son diferentes.

Si tenemos un conjunto de variables del mismo tipo, podemos incluir, en una sola sentencia, todos los identificadores correspondientes separados por comas. Por ejemplo: `int Total, suma, edad;`

T 21 **Operadores**

El lenguaje C permite el uso de una serie de operadores, que de acuerdo al número de operandos sobre los que actúa, se clasifican en unarios (un operando), binarios (dos operandos) y terciarios (tres operandos). A continuación se mencionan algunos de los operadores que se pueden utilizar en C.

a) Operadores Aritméticos

Operador Aritmético	Símbolo usado en C	Ejemplo	Resultado Ejemplo
Suma	+	3+7	10
Resta	-	7-2	5
Multiplicación	*	5*3	15
División entera	/	3/4	0
División	/	7.0/3	2.333333
Módulo (resto de la división)	%	7%2	1
Signo	-	-6	-6

b) Operadores Relacionales

Operador Relacional	Símbolo usado en C	Ejemplo
Mayor	>	a>b
Mayor o igual	>=	a>=b
Menor	<	a<b
Menor o igual	<=	a<=b
Igual que	==	a==b
Distinto que	!=	a!=b

c) Operadores Lógicos

Operador Lógico	Símbolo usado en C	Ejemplo
y (and)	&&	(a>3)&&(b<1)
o (or)		(a==1) (b>=5)
no (not)	!	!(a>b)

d) Operadores de Incremento y Decremento

Operador	Símbolo usado en C	Ejemplos
Incremento	++	i++ ó ++i
Decremento	--	a-- ó --a

e) Operador Condicional

Operador	Símbolos usados en C	Ejemplo
Condicional	? :	(a>=b)?a:b

f) Operadores de Asignación

Operador de Asignación	Símbolo usado en C	Ejemplo
Asignación	=	a=2*b+1
Asignación mixta con suma	+=	a+=b
Asignación mixta con resta	-=	a-=b
Asignación mixta con multiplicación	*=	a*=b
Asignación mixta con división	/=	a/=b
Asignación mixta con módulo	%=	a%=b

▣ 22 **Entrada y salida de datos**

En ▣ 15 se mencionó la función `printf` que permite la salida de datos por pantalla, pero también existe la función `scanf` que permite el ingreso de datos mediante un dispositivo de entrada.

La orden `scanf` es una función de C, que pertenece a la biblioteca *stdio*, que permite leer datos, usando un formato, desde un dispositivo de entrada que habitualmente es el teclado. El usuario, es decir, aquel que utiliza nuestro programa, es quien ingresa la información solicitada mediante `scanf`.

Los especificadores de formato mencionados en ▣ 15 también son válidos para la función `scanf`.

EJERCICIOS

TIPOS DE DATOS, VARIABLES, CONSTANTES, OPERADORES Y, ENTRADA Y SALIDA DE DATOS

- [1] Escriba una aplicación en C que calcule la suma $3 + 6$
- [2] Escriba un programa en C que muestre por pantalla el producto $7 \cdot 3$
- [3] Escriba un programa en C que calcule y muestre por pantalla la suma de dos números enteros ingresados por el usuario

[4] a) Escriba un programa en C que calcule y muestre por pantalla, con 2 decimales, el periodo t de un péndulo de longitud $L = \frac{\sqrt[5]{2}+1}{\sqrt{2}}$ metros.

b) Modifique el programa anterior de modo que el usuario ingrese el valor de la longitud del péndulo. Asuma que la longitud del péndulo es un número real.

(Indicación: $t = 2\pi \sqrt{\frac{L}{g}}$, con $g = 9.81 \text{ m/s}^2$)

[5] Escriba un programa en C que calcule y muestre por pantalla los resultados de las siguientes expresiones:

a) $4 + 6 - 5$

b) $3^5 - \sqrt{6}$

c) $\frac{17 + 4^2}{\sqrt{15} - 1}$

[6] Escriba un programa en C que calcule y muestre por pantalla los resultados de las siguientes expresiones:

a) $3 - 2 - (5 + 7)$

b) $2^3 - 3^2$

c) $\frac{7 + (\sqrt{5} + 4)^2}{\sqrt{15} - 1}$

[7] Use C para obtener y mostrar por pantalla el producto de 3 números reales ingresados por el usuario.

[8] Use C para calcular $f(5)$ si $f(x) = \sqrt{\frac{1}{1+x^3}}$

[9] Escriba un programa en C que permita al usuario convertir grados Fahrenheit a grados Celsius

[10] a) Escriba un programa en C que calcule y muestre por pantalla, con 1 decimal, el promedio simple de tres notas ingresadas por el usuario.
b) Modifique el programa anterior de modo que el usuario también ingrese las ponderaciones de cada nota.

[11] Escriba un programa en C que solicite al usuario el ingreso de dos números reales y muestre por pantalla la suma, la resta, el producto y el cociente del primer número ingresado por el segundo número ingresado. Todos los resultados deben aparecer con 2 decimales.

[12] La distancia d desde la recta $Ax + By + C = 0$ al origen está dada por:

$$d = \frac{|C|}{\sqrt{A^2+B^2}}$$

Escriba un programa en C que muestre por pantalla la distancia desde el origen a la recta $Ax + By + C = 0$ donde A , B y C son ingresados por el usuario. Incluya comentarios apropiados.

[13] Se sabe que el perímetro P de un polígono de n lados circunscrito en una circunferencia de radio real r es $P = 2\pi r \operatorname{tg}\left(\frac{\pi}{n}\right)$

a) Escriba un programa en C que calcule y muestre por pantalla, con 2 decimales, el perímetro de un pentágono circunscrito en una circunferencia de radio igual a $\frac{5}{2}$

b) Modifique el programa anterior de modo que el usuario ingrese los valores del radio y el número de lados.

[14] Escriba un programa en C que solicite al usuario el ingreso de los valores reales a y b , tales que $p(t) = \frac{375}{1 + a e^{-bt}}$, y muestre por pantalla la tabla siguiente :

t	$p(t)$
0	$xxx.xx$
0.5	$xxx.xx$
1	$xxx.xx$

Donde $xxx.xx$ representa el valor de $p(t)$ calculado por el programa, considerando 2 decimales.

Incluya el trazado de las líneas divisorias.

[15] Escriba un programa en C que calcule y muestre por pantalla el promedio de tres notas ingresadas por el usuario.

[16] Escriba un programa en C que calcule y muestre por pantalla el cociente de dos números enteros distintos de cero ingresados por el usuario. El usuario es quien se encarga de ingresar números distintos de cero.

[17] Use C para calcular y mostrar por pantalla el factor de fricción de fanning f para un tipo de fluido newtoniano usando la ecuación de Churchill :

$$f = 2 \left[\left(\frac{8}{Re} \right)^{12} + \frac{1}{(A+B)^{3/2}} \right]^{1/12}$$

con $A = \left[2.457 \cdot \ln \left(\frac{1}{\left(\frac{7}{Re} \right)^{0.9} + 0.27 \cdot \frac{\varepsilon}{d}} \right) \right]^{16}$ y $B = \left(\frac{37530}{Re} \right)^{16}$ donde el número

de Reynolds Re , la rugosidad ε y el diámetro d son números positivos ingresados por el usuario.

Asuma que el usuario introduce valores apropiados.

[18] Escriba un programa en C que calcule y muestre por pantalla la fuerza F de atracción gravitacional entre dos masas m_1 y m_2 situadas a una distancia r , con $F = G \frac{m_1 \cdot m_2}{r^2}$, donde la constante de gravitación universal es igual a $G = 6.67428 \times 10^{-11} \left[\frac{m^3}{kg \cdot s^2} \right]$

El usuario es quien ingresa las masas en kilogramos y la distancia en metros.

[19] a) Escriba un programa en C que calcule y muestre por pantalla, con 1 decimal, el promedio de 5 notas ingresadas por el usuario.

b) Modifique el programa anterior de modo que el usuario ingrese también ponderaciones para cada prueba.

c) ¿Qué cambios tendría el programa original si la primera prueba tiene una ponderación de 10%, la segunda 25% , la tercera 25%, la cuarta 20% y la quinta 20%?

[20] a) Escriba un programa que calcule y muestre por pantalla, con 3 decimales, la distancia entre dos puntos del plano, cuyas componentes son introducidas por el usuario.

b) ¿Qué cambios tendría el programa si queremos calcular la distancia entre dos puntos del espacio?

c) ¿Cómo se podría calcular la distancia desde un punto a una recta en el plano cartesiano?

[21] Escriba un programa en C que calcule y muestre por pantalla, con 3 decimales, el producto interior entre los vectores del plano $\mathbf{a} = (a_1, a_2)$ y $\mathbf{b} = (b_1, b_2)$. Suponga que el usuario es quien ingresa las componentes de los vectores.

[22] Escriba un programa en C que solicite al usuario el ingreso de un ángulo en grados y calcule y muestre por pantalla el seno y el coseno del ángulo ingresado.

[23] Escriba un programa en C que calcule y muestre por pantalla, con 3 decimales, la distancia desde el punto $(1, -1, 4)$ al plano $ax + by + cz = d$, donde a, b, c y d son números reales ingresados por el usuario.

(Indicación:

La distancia entre un punto P_0 y un plano está dada por $\frac{\| \langle \mathbf{n}, P_0 P \rangle \|}{\| \mathbf{n} \|}$, donde P es un punto cualquiera del plano y \mathbf{n} es el vector normal.)

[24] a) Escriba un programa en C que calcule y muestre por pantalla la fuerza F de atracción gravitacional entre dos masas M_1 y M_2 situadas a una distancia R . Las masas y la distancia son ingresadas por el usuario.

Indicación: $F = G \cdot \frac{M_1 \cdot M_2}{R^2}$ newtons, donde las masas se expresan en kilogramos y la distancia en metros, y $G = 6.67428 \times 10^{-11} [N \cdot m^2 / Kg^2]$

b) Reescriba el programa anterior de modo que calcule la fuerza de atracción gravitacional entre La Tierra y La Luna.

[25] Prepare una aplicación en C que solicite al usuario el ingreso de un número entero y otro real, y luego muestre por pantalla el producto de ellos. Use un decimal.

[26] Escriba un programa que calcule el área y el perímetro de un rectángulo de lados 5 y 8.

[27] Escriba un programa en C que pida por pantalla una cantidad en dólares de Estados Unidos y la convierta a pesos chilenos (Asuma que: 1 dólar = 487,04 pesos chilenos).

[28] Escriba un programa en C que convierta de centímetros a pulgadas y viceversa.

El usuario es quien ingresa la información necesaria.

(Indicación: 1 pulgada = 2.54 centímetros)

[29] Escriba un programa en C que muestre por pantalla el área de un triángulo en el espacio cuyos vértices son $A = (x_1, y_1, z_1)$, $B = (x_2, y_2, z_2)$ y $C = (x_3, y_3, z_3)$. Las coordenadas de los vértices son ingresados por el usuario.

(Indicación: El área de un triángulo en el espacio es :

$$Area = \frac{1}{2} \| \overline{AB} \times \overline{BC} \|)$$

[30] Escriba un programa en C que solicite al usuario el ingreso de un cierto número de segundos y muestre por pantalla el número de horas, minutos y segundos asociados al dato ingresado por el usuario.

[31] Escriba un programa en C que calcule y muestre por pantalla la suma de los restos de las divisiones $5/4$ y $3/8$

[32] Escriba un programa en C que calcule y muestre por pantalla, con 3 decimales, la distancia desde el punto $(1, -1, 4)$ al plano $ax + by + cz = d$, donde a, b, c y d son números reales ingresados por el usuario.

[33] Escriba un programa en C que calcule y muestre por pantalla el valor de la función $f(x) = \frac{\sqrt[5]{x} e^{x^3}}{x^2+1} + \sqrt{\pi}$ donde x es un valor entero ingresado por el usuario.

[34] Use C para calcular y mostrar por pantalla, en color azul brillante, el resultado de $\frac{4 \times 10^4}{7 \times 10^{-7}} \cdot \frac{\text{sen}^3(x)}{1 + \text{tg}(x)}$, donde x es un valor entero, en grados, ingresado por el usuario. Use notación científica para expresar el resultado.

[35] Escriba una aplicación en C que calcule y muestre por pantalla $p(a)$, donde a es ingresado por el usuario y $p(x) = 3x^2 - 2x + \sqrt{3}$

[36] Use C para calcular y mostrar por pantalla el resultado de $\frac{4 \times 10^6}{3 \times 10^n} + \ln(|\sqrt[5]{n^2}|)$, usando notación científica, donde n es un número entero ingresado por el usuario.

[37] Escriba un programa en C que calcule y muestre por pantalla el valor de $\frac{\cos(\sqrt{2}\pi) + (-1)^5}{1 + \sqrt[3]{9}}$

[38] Escriba un programa en C que calcule y muestre por pantalla el valor $f(4)$ para $f(x) = \frac{1-x^2}{1+x^3}$

[39] Escriba un programa en C que muestre por pantalla, usando color verde brillante, la siguiente tabla :

x	x^2
2	4
6	36

Los cuadrados de los valores de x deben ser calculados por el programa.

[40] Escriba un programa en C que calcule y muestre por pantalla, en color verde brillante, el producto $(a + b)(c + d)$, donde a, b, c y d son números enteros ingresados por el usuario.

[41] Escriba un programa en C que muestre por pantalla, usando color rojo brillante, la siguiente tabla :

n	A
3	$xx.x$
4	$xx.x$

Observe que $xx.x$ representa los valores de $A = \frac{1}{2} n \operatorname{sen}\left(\frac{2\pi}{n}\right)$ con un decimal, para el valor correspondiente de n .

Incluya el trazado de las líneas divisorias.

[42] Escriba una aplicación en C que calcule y muestre por pantalla el resultado de $\frac{3 \cos^2(x)}{1 + \operatorname{tg}(x)}$ para $x = 47^\circ$

[43] Prepare una aplicación en C que muestre por pantalla el resultado de $\frac{3 \times 10^{-4}}{6 \times 10^5}$ usando notación científica.

[44] Use C para solicitar al usuario el ingreso de las coordenadas (x, y) de un punto del plano y luego convertir tales coordenadas cartesianas a polares.

[45] Use C para calcular la suma, la resta, el producto y el cociente de dos números complejos ingresados por el usuario. Considere dos decimales.

[46] Escriba un programa en C que muestre la fecha y hora actual por pantalla.

[47] Escriba un programa en C que permita calcular el volumen de un cono de altura 3 cm. y radio basal 1.5 cm.

$$(Ind. : V_{cono} = \frac{1}{3} S h, \text{ con } S = \pi r^2)$$

[48] Escriba un programa en C que muestre por pantalla los valores de la función

$$f(x) = \begin{cases} x^3 - 3 & , \text{ si } x < -4 \\ \sqrt{x+4} & , \text{ si } -4 \leq x < 4 \\ \cos\left(\frac{1}{x+4}\pi\right) & , \text{ si } x \geq 4 \end{cases}$$

en $x = -6$, en $x = 1$ y en $x = 2\pi$.

[49] Se sabe que el área A de un triángulo cuyos lados son los números enteros a , b y c , y donde $s = \frac{a+b+c}{2}$ está dada por $A = \sqrt{s(s-a)(s-b)(s-c)}$

a) Escriba un programa en C que calcule y muestre por pantalla el área de un triángulo cuyos lados tienen longitudes $a = 4$, $b = 2$ y $c = 3$

b) Escriba un programa en C que calcule y muestre por pantalla el área de un triángulo tal que las longitudes de los lados son ingresadas por el usuario.

PROBLEMAS RESUELTOS

TIPOS DE DATOS, VARIABLES, CONSTANTES, OPERADORES Y, ENTRADA Y SALIDA DE DATOS

1

Escriba una aplicación en C que calcule la suma $3 + 6$

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main()
{
    printf("\n%i",3+6);
    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida

```
9
Presione una tecla para continuar . . .
```

Observaciones:

- 1) Notamos que la orden printf permite mostrar el resultado de una operación, en este caso, la suma $3 + 6$.
- 2) En el ejercicio usamos el especificador de formato %i para indicar que la suma $3 + 6$ da como resultado un número entero. Si hubiésemos escrito %f en vez de %i el resultado sería

```
0.000000
Presione una tecla para continuar . . .
```

lo que nos indica que existe incompatibilidad entre el especificador de formato usado y el resultado de la operación; recordemos que la suma de dos números enteros da como resultado otro número entero.

Muy distinto es el resultado, si dejamos en claro que no estamos sumando dos números enteros, sino dos números reales, lo que podría hacerse del siguiente modo:

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    printf("\n%f",3.0+6.0);
    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

La salida del programa anterior es:

```
9.000000
Presione una tecla para continuar . . .
```

que muestra el resultado correcto, pues al escribir 3.0+6.0 dejamos en claro que la suma es de dos números reales.

Ahora, si sumo un número entero con uno real el resultado es real, como se muestra en el siguiente programa

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    printf("\n%f",3+6.0);
}
```

```
printf("\n\n");
system("PAUSE");
return 0;
}
```

cuya salida es

```
9.000000
Presione una tecla para continuar . . .
```

3) Se observa que C muestra, por defecto, un resultado real usando 6 decimales.

3

Escriba un programa en C que calcule y muestre por pantalla la suma de dos números enteros ingresados por el usuario

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int num1,num2, suma;
    scanf("%i",&num1);
    scanf("%i",&num2);

    suma=num1+num2;

    printf("La suma es %i",suma);

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida

```
3
2
La suma es 5
Presione una tecla para continuar . . .
```

Observaciones:

1) La sentencia `int num1,num2, suma;` declara tres variables (`num1,num2` y `suma`) del mismo tipo (`int`) en una sola línea de código.

2) En el programa anterior, al momento de la ejecución, el cursor queda parpadeando a la espera de que el usuario ingrese la información necesaria, lo cual puede ser confuso, por lo que siempre es apropiado indicarle al usuario lo que está ingresando, de donde sería muy útil incluir mensajes adecuados antes de cada `scanf` para decirle al usuario qué es lo que debe ingresar. Tomando en cuenta esta consideración el programa podría ser:

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int num1,num2, suma;
    printf("\nIngrese primer n%cmero entero : ",163);
    scanf("%i",&num1);
    printf("\nIngrese segundo n%cmero entero : ",163);
    scanf("%i",&num2);

    suma=num1+num2;

    printf("\nLa suma es %i",suma);

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

cuya salida es:

```
Ingrese primer número entero : 5
Ingrese segundo número entero : 3
La suma es 8
Presione una tecla para continuar . . .
```

3) En la aplicación anterior no es necesario el uso de la variable *suma*, pudiendo omitirse de la siguiente manera:

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int num1,num2;

    printf("\nIngrese primer número entero : ",163);
    scanf("%i",&num1);
    printf("\nIngrese segundo número entero : ",163);
    scanf("%i",&num2);

    printf("\nLa suma es %i",num1+num2);

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

La salida de este programa coincide exactamente con la de la observación anterior.

4

a) Escriba un programa en C que calcule y muestre por pantalla, con 2 decimales, el periodo t de un péndulo de longitud $L = \frac{\sqrt[5]{2}+1}{\sqrt{2}}$ metros.

b) Modifique el programa anterior de modo que el usuario ingrese el valor de la longitud del péndulo. Asuma que la longitud del péndulo es un número real.

(Indicación: $t = 2\pi\sqrt{\frac{L}{g}}$, con $g = 9.81\text{ m/s}^2$)

Solución:

a)

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define g 9.81

int main()
{
    float L=(pow(2,1.0/5)+1)/sqrt(2);

    printf("\nEl periodo es %0.2f\n\n",2*M_PI*sqrt(L/g));

    system("PAUSE");
    return 0;
}
```

Salida

```
El periodo es 2.47
Presione una tecla para continuar . . .
```

Observaciones:

1) En este ejercicio debemos incluir la librería math pues hacemos uso de las funciones *pow* y *sqrt*, y la constante *M_PI*, que pertenecen a tal librería.

La función `pow(base,exp)` permite calcular la potencia $base^{exp}$, por ejemplo, `pow(2,1.0/5)` permite calcular $\sqrt[5]{2} = 2^{1/5}$

La función `sqrt(x)` calcula la raíz cuadrada de x

La constante `M_PI` es igual a 3.14159265358979323846, que es aproximadamente igual a π .

Notemos que en la expresión `pow(2,1.0/5)` es muy importante indicarle a C que el exponente $1/5$ es un número real, para ello se incluye el punto después del 1. Si hubiésemos escrito `pow(2,1/5)` entonces C toma el cociente $1/5$ como una división de enteros y el resultado de tal división es la parte entera del cociente, es decir, 0; por lo que el resultado final de `pow(2,1/5)` es en este caso igual a 1.

2) La directiva de preprocesado `#define x y` permite reemplazar x por y en el programa, antes del proceso de compilación.

3) Otra forma de resolver el ejercicio, sin considerar la directiva de preprocesado `#define`, es de la siguiente manera:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    const float g=9.81;

    float L=(pow(2,1.0/5)+1)/sqrt(2);

    printf("\nEl periodo es %0.2f\n\n",2*M_PI*sqrt(L/g));

    system("PAUSE");
    return 0;
}
```

En este caso se hace uso del cualificador `const` que permite declarar una variable de algún tipo como una constante, es decir, el valor de tal variable no cambiará durante la ejecución del programa.

La salida, en este caso, es exactamente igual a la del programa anterior.

4) Otra forma de resolver el problema es:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    float g=9.81;

    float L=(pow(2,1.0/5)+1)/sqrt(2);

    printf("\nEl periodo es %0.2f\n\n",2*M_PI*sqrt(L/g));

    system("PAUSE");
    return 0;
}
```

Este programa, a diferencia del anterior, considera que g es una variable cuyo valor inicial es 9.81

La salida, en este caso, es exactamente igual a la de los programas anteriores.

b)

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    float L, t;
    const float g=9.81;

    printf("\nIngrese la longitud :");
    scanf("%f",&L);

    t=2*M_PI*sqrt(L/g);

    printf("\nEl periodo de un p%cnulo de longitud %0.2f es %0.2f\n\n",130,L,t);
```

```
system("PAUSE");
return 0;
}
```

Salida

```
Ingrese la longitud :3.56
El periodo de un péndulo de longitud 3.56 es 3.79
Presione una tecla para continuar . . .
```

Observaciones:

1) Aunque es innecesario, en la solución anterior hemos agregado una variable t que almacena el valor del periodo.

2) La línea `t=2*M_PI*sqrt(L/g);` permite asignar a la variable t el valor de $2 * M_PI * \sqrt{L/g}$, donde L es ingresado por el usuario y g es una constante de tipo real declarada al inicio del programa principal.

6

Escriba un programa en C que calcule y muestre por pantalla los resultados de las siguientes expresiones:

a) $3 - 2 - (5 + 7)$

b) $2^3 - 3^2$

c) $\frac{7 + (\sqrt{5} + 4)^2}{\sqrt{15} - 1}$

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
```

```

int main()
{
    printf("a) %i\n",3-2-(5+7));
    printf("b) %f\n",pow(2,3)-pow(3,2));
    printf("c) %f\n", (7+pow(sqrt(5)+4,2))/(sqrt(15)-1));
    system("PAUSE");
    return 0;
}

```

Salida

```

a> -11
b> -1.000000
c> 15.972436
Presione una tecla para continuar . . .

```

Observaciones:

1) Notemos que la función pow entrega como resultado un valor real, y aunque $2^3 - 3^2 = 8 - 9 = -1$ da como resultado un entero debemos obligatoriamente usar el especificador de formato %f, porque si usamos %i el resultado será 0

2) Observemos que $(\sqrt{5} + 4)^2$ se escribe en C como pow(sqrt(5)+4,2), o también se podría escribir (sqrt(5)+4)*(sqrt(5)+4)

8

Use C para calcular $f(5)$ si $f(x) = \sqrt{\frac{1}{1+x^3}}$

Solución:

Código Fuente

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

```

```

int main()

```

```

{
  int x=5;
  float fx;
  fx=sqrt(1./(1+x*x*x));
  printf("\nf(%i) = %f\n\n",x,fx);
  system("PAUSE");
  return 0;
}

```

Salida

```

f<5> = 0.089087
Presione una tecla para continuar . . .

```

Observaciones:

- 1) En vez de usar la función pow hemos usado el hecho que $x^3 = x*x*x$
- 2) Al calcular $f(x) = \sqrt{\frac{1}{1+x^3}}$ es importante dejar en claro que el cuociente $\frac{1}{1+x^3}$ debe entregar como resultado un valor real, para ello, se incluye el punto después del 1 en el cuociente $1./(1+x*x*x)$
Note que 1. es lo mismo que 1.0
- 3) Otra forma alternativa, sin incluir el punto después del 1, de resolver el ejercicio es considerando que el argumento x de la función f es un valor real, en cuyo caso el programa queda como:

```

#include <stdio.h>
#include <stdlib.h>

int main()
{
  float x=5;
  float fx;
  fx=sqrt(1/(1+x*x*x));
  printf("\nf(%f) = %f\n\n",x,fx);
  system("PAUSE");
  return 0;
}

```

y la salida es:

```
f<5.000000> = 0.009007  
Presione una tecla para continuar . . .
```



9

Escriba un programa en C que permita al usuario convertir grados Fahrenheit a grados Celsius

Solución:

Código Fuente

```
#include <stdio.h>  
#include <stdlib.h>  
  
int main()  
{  
    float F, C;  
    printf("\nIngrese la temperatura en grados Fahrenheit :");  
    scanf("%f",&F);  
    C=5.0/9*(F-32);  
    printf("\n%0.1f°F es igual a %0.1f°C\n\n",F,248,C,248);  
  
    system("PAUSE");  
    return 0;  
}
```

Salida

```
Ingrese la temperatura en grados Fahrenheit :90  
90.0°F es igual a 32.2°C  
Presione una tecla para continuar . . .
```

Observaciones:

1) La fórmula de conversión de grados Fahrenheit (F) a Celsius (C) es:

$$C = \frac{5}{9}(F - 32)$$

2) El código ASCII para ° es 248

3) Los resultados se muestran con 1 decimal gracias al especificador de formato %0.1f

12

La distancia d desde la recta $Ax + By + C = 0$ al origen está dada por:

$$d = \frac{|C|}{\sqrt{A^2 + B^2}}$$

Escriba un programa en C que muestre por pantalla la distancia desde el origen a la recta $Ax + By + C = 0$ donde A , B y C son ingresados por el usuario. Incluya comentarios apropiados.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    // Declaración de variables
    float A, B, C;
    float d;

    /* Ingreso de los coeficientes de la recta */
    printf("\nIngrese el valor de A : ");
    scanf("%f",&A);
    printf("\nIngrese el valor de B : ");
    scanf("%f",&B);
```

```

printf("\nIngrese el valor de C : ");
scanf("%f",&C);

/* Cálculo de la distancia */
d=fabs(C)/sqrt(A*A+B*B); //fabs permite calcular el valor absoluto

/* Se muestra el valor de la distancia por pantalla */
printf("\nLa distancia es %f\n\n",d);

/* Pausa para observar la salida del programa */
system("PAUSE");

/* Valor de retorno de la función principal ( main() ),
si la ejecución ha sido exitosa */
return 0;
}

```

Salida

```

Ingrese el valor de A : 3
Ingrese el valor de B : 2
Ingrese el valor de C : -1
La distancia es 0.277350
Presione una tecla para continuar . . .

```

Observaciones:

1) Siempre es posible y recomendable incluir comentarios en el código fuente, pues sirven para que el programador recuerde lo que hizo en el momento de crear un programa y también para que otros programadores sepan para qué sirve cada sección de código.

Recordemos que los comentarios son ignorados por los compiladores.

2) La función $\text{fabs}(x)$, presente en la biblioteca `math.h`, permite calcular $|x|$, donde x es una variable de tipo real (float).

También existe la función $\text{abs}(x)$, presente en la biblioteca `stdlib.h`, que también calcula el valor absoluto, pero en este caso de un argumento x entero (int).

14

Escriba un programa en C que solicite al usuario el ingreso de los valores reales a y b , tales que $p(t) = \frac{375}{1 + a e^{-bt}}$, y muestre por pantalla la tabla siguiente:

t	$p(t)$
0	$xxx.xx$
0.5	$xxx.xx$
1	$xxx.xx$

Donde $xxx.xx$ representa el valor de $p(t)$ calculado por el programa, considerando 2 decimales.

Incluya el trazado de las líneas divisorias.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    float a,b;
    float pt;
    printf("\nIngrese a : ");
    scanf("%f",&a);
    printf("\nIngrese b : ");
    scanf("%f",&b);

    printf("\n%c%c%c%c%c%c%c%c",218,196,196,196,196,196,196);
    printf("%c%c%c%c%c%c%c%c%c%c",194,196,196,196,196,196,196,196,191);

    printf("\n%c t   %c",179,179);
    printf(" p(t) %c",179);

    printf("\n%c%c%c%c%c%c%c%c",195,196,196,196,196,196,196);
    printf("%c%c%c%c%c%c%c%c%c%c",197,196,196,196,196,196,196,180);
```

```

printf("\n%c 0  %c",179,179);
printf(" %6.2f %c",375/(1+a*exp(-b*0)),179);
printf("\n%c 0.5  %c",179,179);
printf(" %6.2f %c",375/(1+a*exp(-b*0.5)),179);
printf("\n%c 1  %c",179,179);
printf(" %6.2f %c",375/(1+a*exp(-b*1)),179);

printf("\n%c%c%c%c%c%c%c%c",192,196,196,196,196,196,196);
printf("%c%c%c%c%c%c%c%c%c%c",193,196,196,196,196,196,196,196,196,217);

printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida

```

Ingreso a : 2
Ingreso b : 1

```

t	p(t)
0	125.00
0.5	169.45
1	216.04

```

Presione una tecla para continuar . . .

```

Observaciones:

1) La función $\exp(x)$, perteneciente a la biblioteca `math.h`, permite calcular la exponencial e^x , donde x es un valor real.

2) En este ejercicio es necesario el uso del especificador de formato `%6.2f` que permite encuadrar perfectamente los resultados de la función $p(t)$ evaluada en $t = 0$, $t = 0.5$ y $t = 1$.

Note que `%6.2f` indica que el valor asociado a este especificador de formato ocupa un total de 6 caracteres, de los cuales 2 son para decimales. El punto decimal también se cuenta.

17

Use C para calcular y mostrar por pantalla el factor de fricción de fanning f para un tipo de fluido newtoniano usando la ecuación de Churchill :

$$f = 2 \left[\left(\frac{8}{Re} \right)^{12} + \frac{1}{(A+B)^{3/2}} \right]^{1/12}$$

con $A = \left[2.457 \cdot \ln \left(\frac{1}{\left(\frac{7}{Re} \right)^{0.9} + 0.27 \cdot \frac{\varepsilon}{d}} \right) \right]^{16}$ y $B = \left(\frac{37530}{Re} \right)^{16}$ donde el

número de Reynolds Re , la rugosidad ε y el diámetro d son números positivos ingresados por el usuario.

Asuma que el usuario introduce valores apropiados.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    float Re, e, d, f, A, B;
    printf("\nIngrese el n°mero de Reynolds : ",163);
    scanf("%f",&Re);

    printf("\nIngrese la rugosidad : ");
    scanf("%f",&e);

    printf("\nIngrese el di°metro : ",160);
    scanf("%f",&d);

    A=pow(2.457*log(1/(pow(7.0/Re,0.9)+0.27*e/d)),16);
    B=pow(37530.0/Re,16);
    f=2*pow(pow(8.0/Re,12)+1/pow(A+B,3.0/2.0),1.0/12.0);

    printf("\nEl factor de fricci°cn de fanning es %0.3f\n\n",162,f);
```

```
system("PAUSE");
return 0;
}
```

Salida

```
Ingrese el número de Reynolds : 6500
Ingrese la rugosidad : 0.8
Ingrese el diámetro : 19
El factor de fricción de fanning es 0.017876
Presione una tecla para continuar . . .
```

Observaciones:

- 1) La función $\log(x)$, perteneciente a la biblioteca `math.h`, permite calcular el logaritmo natural $\ln(x)$, donde x es un valor real.
- 2) Para calcular f , se ha hecho uso de la función `pow` como argumento de otra función `pow`, lo que es perfectamente lícito, y se denomina funciones anidadas.
- 3) El símbolo ε usado en el enunciado para la rugosidad, no es posible incorporarlo como tal en el programa, porque no se pueden usar tales símbolos como identificadores válidos. Por tal motivo, a la rugosidad se le llamó `e` en el código fuente.

18

Escriba un programa en C que calcule y muestre por pantalla la fuerza F de atracción gravitacional entre dos masas m_1 y m_2 situadas a una distancia r , con $F = G \frac{m_1 \cdot m_2}{r^2}$, donde la constante de gravitación universal es igual a $G = 6.67428 \times 10^{-11} \left[\frac{m^3}{kg \cdot s^2} \right]$. El usuario es quien ingresa las masas en kilogramos y la distancia en metros.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    float G=6.67428e-11;
    float m1,m2,r,F;

    printf("\nIngrese la masa 1 (kg): ");
    scanf("%f",&m1);

    printf("\nIngrese la masa 2 (kg) : ");
    scanf("%f",&m2);

    printf("\nIngrese la distancia entre ellas (m): ");
    scanf("%f",&r);

    F=G*(m1*m2)/pow(r,2);

    printf("\nLa fuerza de atracci%cn gravitacional es %e",162,F);
    printf("\n\n");

    system("PAUSE");
    return 0;
}
```

Salida

```
Ingrese la masa 1 <kg>: 500
Ingrese la masa 2 <kg> : 5000
Ingrese la distancia entre ellas <m>: 450000
La fuerza de atracción gravitacional es 8.239852e-016
Presione una tecla para continuar . . .
```

Observaciones:

- 1) Es importante, en este ejercicio, el uso del especificador de formato %e para mostrar el resultado por pantalla en notación científica. Si usamos %f en vez de %e la salida sería:

```
Ingrese la masa 1 (kg): 500
Ingrese la masa 2 (kg) : 5000
Ingrese la distancia entre ellas (m): 450000
La fuerza de atracción gravitacional es 0.000000
Presione una tecla para continuar . . .
```

Esto podría llevarnos a la conclusión errónea de que la fuerza gravitacional entre las masas es igual a cero, pero eso no es cierto, como se observa en la salida usando %e

- 2) A la constante de gravitación universal se le asigna su valor usando también notación científica.
- 3) También es posible, en C, ingresar los valores usando notación científica, como se muestra en la siguientes salidas

Salida 1

```
Ingrese la masa 1 (kg): 5e2
Ingrese la masa 2 (kg) : 5e3
Ingrese la distancia entre ellas (m): 45e4
La fuerza de atracción gravitacional es 8.239852e-016
Presione una tecla para continuar . . .
```

Salida 2

```
Ingrese la masa 1 <kg>: 50
Ingrese la masa 2 <kg> : 5.974e24
Ingrese la distancia entre ellas <m>: 6378140
La fuerza de atracción gravitacional es 4.900626e+002
Presione una tecla para continuar . . .
```



20

- Escriba un programa que calcule y muestre por pantalla, con 3 decimales, la distancia entre dos puntos del plano, cuyas componentes son introducidas por el usuario.
- ¿Qué cambios tendría el programa si queremos calcular la distancia entre dos puntos del espacio?
- ¿Cómo se podría calcular la distancia desde un punto a una recta en el plano cartesiano?

Solución:

a)

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main()
```

```
{
```

```
float x1, y1, x2, y2, Distancia;
```

```
printf("\nIngrese las coordenadas del primer punto\n");
```

```
printf("\n x1 : ");
```

```
scanf("%f",&x1);
```

```

printf("\n y1 : ");
scanf("%f",&y1);

printf("\n");

printf("\nIngrese las coordenadas del segundo punto\n");
printf("\n x2 : ");
scanf("%f",&x2);

printf("\n y2 : ");
scanf("%f",&y2);

Distancia = sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2));

printf("\nLa distancia es %0.3f",Distancia);

printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida

```

Ingrese las coordenadas del primer punto
x1 : -1
y1 : -1

Ingrese las coordenadas del segundo punto
x2 : 2
y2 : 4

La distancia es 5.831
Presione una tecla para continuar . . .

```

Observaciones:

1) La distancia entre los puntos del plano (x_1, y_1) y (x_2, y_2) es igual a

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

2) No es posible incluir subíndices en un nombre de variable

3) El lenguaje C hace diferencia entre mayúsculas y minúsculas, es decir, Distancia es un nombre de variable distinto a distancia.

Si en el programa declaramos distancia usando minúscula y luego al calcularla usamos mayúscula, se obtiene un mensaje de error indicando que Distancia es una variable que aún no ha sido declarada, lo que concuerda con lo mencionado, pues se ha declarado la variable distancia que usa minúscula y no Distancia que usa mayúscula.

b)

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main()
{
    float x1, y1, x2, y2, z1, z2, Distancia;

    printf("\nIngrese las coordenadas del primer punto\n");
    printf("\n x1 : ");
    scanf("%f",&x1);

    printf("\n y1 : ");
    scanf("%f",&y1);

    printf("\n z1 : ");
    scanf("%f",&z1);

    printf("\n");

    printf("\nIngrese las coordenadas del segundo punto\n");
    printf("\n x2 : ");
    scanf("%f",&x2);

    printf("\n y2 : ");
    scanf("%f",&y2);

    printf("\n z2 : ");
    scanf("%f",&z2);
```

```

Distancia = sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2)+(z1-z2)*(z1-z2));

printf("\nLa distancia es %0.3f",Distancia);

printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida

```

Ingrese las coordenadas del primer punto
x1 : 1
y1 : -1
z1 : 3

Ingrese las coordenadas del segundo punto
x2 : 2
y2 : 1
z2 : -4

La distancia es 7.348
Presione una tecla para continuar . . .

```

Observación:

La distancia entre los puntos del espacio (x_1, y_1, z_1) y (x_2, y_2, z_2) es igual a

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \quad \square$$

c)

Código Fuente

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

```

```

int main()
{
    float x0, y0, A, B, C, d;

    printf("\nIngrese las coordenadas del punto\n");
    printf("\n x0 : ");
    scanf("%f",&x0);

    printf("\n y0 : ");
    scanf("%f",&y0);

    printf("\n");

    printf("\nIngrese los coeficientes de la recta\n");
    printf("\n A : ");
    scanf("%f",&A);

    printf("\n B : ");
    scanf("%f",&B);

    printf("\n C : ");
    scanf("%f",&C);

    d = fabs(A*x0+B*y0+C)/sqrt(A*A+B*B);

    printf("\nLa distancia es %0.3f",d);

    printf("\n\n");
    system("PAUSE");
    return 0;
}

```

Salida

```
Ingrese las coordenadas del punto
x0 : 2
y0 : -1

Ingrese los coeficientes de la recta
A : 3
B : 4
C : 0

La distancia es 0.400
Presione una tecla para continuar . . .
```

Observación:

La distancia d del punto (x_0, y_0) a la recta de ecuación $Ax + By + C = 0$ es igual a

$$d = \frac{|Ax_0 + By_0 + C|}{\sqrt{A^2 + B^2}} \quad \square$$

22

Escriba un programa en C que solicite al usuario el ingreso de un ángulo en grados y calcule y muestre por pantalla el seno y el coseno del ángulo ingresado.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
```

```

int main()
{
    float ang, rad;

    printf("\nIngrese el ángulo en grados : ",160);
    scanf("%f",&ang);

    rad=M_PI/180*ang;

    printf("\nsen(%0.1f%c) = %0.5f",ang,248,sin(rad));
    printf("\n\ncos(%0.1f%c) = %0.5f",ang,248,cos(rad));

    printf("\n\n");
    system("PAUSE");
    return 0;
}

```

Salida

```

Ingrese el ángulo en grados : 30
sen(30.0°) = 0.50000
cos(30.0°) = 0.86603
Presione una tecla para continuar . . .

```

Observaciones:

1) El lenguaje C trabaja usando radianes, por lo que es necesaria la conversión a radianes cuando corresponda.

Para convertir el ángulo α en grados a radianes podemos usar la regla:

$$\frac{\alpha \text{ radianes}}{\pi} = \frac{\alpha \text{ grados}}{180}$$

De donde se obtiene que: $\alpha \text{ radianes} = \frac{\pi}{180} \alpha \text{ grados}$

2) Las funciones $\sin(x)$ y $\cos(x)$ incluidas en la biblioteca `math.h`, permiten calcular el seno y el coseno, respectivamente, del ángulo x expresado en radianes. \square

25

Prepare una aplicación en C que solicite al usuario el ingreso de un número entero y otro real, y luego muestre por pantalla el producto de ellos. Use un decimal.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int num1; float num2;

    printf("\nIngrese un número entero : ");
    scanf("%d",&num1);
    printf("\nIngrese un número real : ");
    scanf("%f",&num2);

    printf("\n\t%i * %0.1f = %0.1f",num1,num2,num1*num2);

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida

```
Ingrese un número entero : 4
Ingrese un número real : 2.6
      4 * 2.6 = 10.4
Presione una tecla para continuar . . .
```

Observaciones:

1) El producto de un número entero por uno real da como resultado un número real, es por esto que el especificador de formato del producto es %0.1f

2) Es posible incluir más de una instrucción simple o elemental en una sola línea de programa como por ejemplo: `int num1; float num2;`

Cada sentencia simple o elemental debe ir separada por un punto y coma (;)

3) Si al ingresar el número entero cometemos un error e ingresamos un número real, la ejecución es anómala, como aparece en la siguiente salida

```
Ingrese un número entero : 5.6
Ingrese un número real :
    5 * 0.6 = 3.0
Presione una tecla para continuar . . .
```

En la salida anterior, se le permitió al usuario ingresar sólo el primer número y de forma inmediata se exhibió el resultado, sin posibilidades de ingresar el segundo número. Además, notamos que se multiplican la parte entera del número ingresado con su parte decimal.

28

Escriba un programa en C que convierta de centímetros a pulgadas y viceversa. El usuario es quien ingresa la información necesaria.
(Indicación: 1 pulgada = 2.54 centímetros)

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main()
{
    float cm, pulg;
    printf("\nConversi%cn de cent%cmetros a pulgadas\n",162,161);
    printf("\nIngrese la cantidad en cm : ");
    scanf("%f",&cm);
```

```

pulg=cm/2.54;
printf("\n%0.2f cm = %0.2f pulgadas\n",cm,pulg);

printf("\n\nConversi%cn de pulgadas a cent%cmetros\n",162,161);
printf("\nIngrese la cantidad en pulgadas : ");
scanf("%f",&pulg);
cm=2.54*pulg;
printf("\n%0.2f pulgadas = %0.2f cm\n",pulg,cm);

printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida

```

Conversión de centímetros a pulgadas
Ingrese la cantidad en cm : 12.78
12.78 cm = 5.03 pulgadas

Conversión de pulgadas a centímetros
Ingrese la cantidad en pulgadas : 13.5
13.50 pulgadas = 34.29 cm

Presione una tecla para continuar . . .

```

Observaciones:

1) Para realizar la conversión podemos usar la expresión:

$$\frac{1 \text{ pulgada}}{x \text{ pulgada}} = \frac{2.54 \text{ cm}}{y \text{ cm}}$$

De donde:

$$x \text{ pulgada} = \frac{y \text{ cm}}{2.54}$$

$$y = 2.54 \cdot x \text{ pulgadas}$$

2) En el código fuente se han usado sólo dos variables para trabajar ambas conversiones. Esto es posible porque, por ejemplo, según la salida anterior la variable cm tiene un valor inicial dado por el usuario igual a 12.78, luego la variable pulg almacena el valor 5.03 dado por la fórmula correspondiente y justo en este punto se exhibe por pantalla el valor 5.03; enseguida la variable pulg toma otro valor dado por el que ingresa el usuario, es decir, 13.5 para finalmente la variable cm almacenar el valor 34.29 que es calculado con la fórmula correspondiente y luego mostrado por pantalla.

29

Escriba un programa en C que muestre por pantalla el área de un triángulo en el espacio cuyos vértices son $A = (x_1, y_1, z_1)$, $B = (x_2, y_2, z_2)$ y $C = (x_3, y_3, z_3)$. Las coordenadas de los vértices son ingresados por el usuario.
(Indicación: El área de un triángulo en el espacio es :

$$Area = \frac{1}{2} \| \overline{AB} \times \overline{BC} \|$$

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    float x1,y1,z1,x2,y2,z2,x3,y3,z3;
    float X21, X32, Y21, Y32, Z21, Z32;
    float dif1, dif2, dif3, Area;
    printf("\nCoordenadas del primer v%crtrice\n",130);
    printf("\n x1 : ");
    scanf("%f",&x1);
    printf("\n y1 : ");
```

```

scanf("%f",&y1);
printf("\n z1 : ");
scanf("%f",&z1);

printf("\nCoordenadas del segundo v%rtice\n",130);
printf("\n x2 : ");
scanf("%f",&x2);
printf("\n y2 : ");
scanf("%f",&y2);
printf("\n z2 : ");
scanf("%f",&z2);

printf("\nCoordenadas del tercer v%rtice\n",130);
printf("\n x3 : ");
scanf("%f",&x3);
printf("\n y3 : ");
scanf("%f",&y3);
printf("\n z3 : ");
scanf("%f",&z3);

X21=x2-x1; X32=x3-x2;
Y21=y2-y1; Y32=y3-y2;
Z21=z2-z1; Z32=z3-z2;
dif1=Y21*Z32-Z21*Y32;
dif2=X32*Z21-X21*Z32;
dif3=X21*Y32-Y21*X32;

Area=1./2*sqrt(dif1*dif1+dif2*dif2+dif3*dif3);

printf("\nEl %crea del tri%cngulo en el espacio es : %0.2f",160,160,Area);

printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida

```
Coordenadas del primer vértice
x1 : 1
y1 : -2
z1 : 3
Coordenadas del segundo vértice
x2 : 1
y2 : 2
z2 : 5
Coordenadas del tercer vértice
x3 : -3
y3 : -2
z3 : 4
El área del triángulo en el espacio es : 9.17
Presione una tecla para continuar . . .
```

Observaciones:

1) El área del triángulo de vértices $A = (x_1, y_1, z_1)$, $B = (x_2, y_2, z_2)$ y $C = (x_3, y_3, z_3)$ está dado por:

$$Area = \frac{1}{2} \|\overline{AB} \times \overline{BC}\| =$$

$$\frac{1}{2} \|[x_2 - x_1, y_2 - y_1, z_2 - z_1] \times [x_3 - x_2, y_3 - y_2, z_3 - z_2]\| =$$

$$\frac{1}{2} \left\| \begin{array}{ccc} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_2 & y_3 - y_2 & z_3 - z_2 \end{array} \right\| =$$

$$\frac{1}{2} \|[Y_{21} \cdot Z_{32} - Z_{21} \cdot Y_{32}, X_{32} \cdot Z_{21} - X_{21} \cdot Z_{32}, X_{21} \cdot Y_{32} - Y_{21} \cdot X_{32}]\| =$$

$$\frac{1}{2} \sqrt{[dif1]^2 + [dif2]^2 + [dif3]^2}$$

con

$Y_{21} = y_2 - y_1$, $Z_{32} = z_3 - z_2$, $Z_{21} = z_2 - z_1$, $Y_{32} = y_3 - y_2$, $X_{32} = x_3 - x_2$,
 $X_{21} = x_2 - x_1$, $dif1 = Y_{21} \cdot Z_{32} - Z_{21} \cdot Y_{32}$, $dif2 = X_{32} \cdot Z_{21} - X_{21} \cdot Z_{32}$ y
 $dif3 = X_{21} \cdot Y_{32} - Y_{21} \cdot X_{32}$

2) Si los puntos ingresados están en una misma recta, entonces el área del triángulo es cero, como se muestra en la siguiente salida

```
Coordenadas del primer vértice
x1 : 1
y1 : 1
z1 : 1
Coordenadas del segundo vértice
x2 : 2
y2 : 2
z2 : 2
Coordenadas del tercer vértice
x3 : 3
y3 : 3
z3 : 3
El área del triángulo en el espacio es : 0.00
Presione una tecla para continuar . . .
```



30

Escriba un programa en C que solicite al usuario el ingreso de un cierto número de segundos y muestre por pantalla el número de horas, minutos y segundos asociados al dato ingresado por el usuario.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main()
```

```

{

int NumSegundos, NumHoras, NumMinutos, NumSegundos2;

printf("\nIngrese un cierto número de segundos :",163);
scanf("%d",&NumSegundos);

NumHoras=NumSegundos/3600;
NumMinutos=(NumSegundos-(NumHoras*3600))/60;
NumSegundos2=(NumSegundos-(NumHoras*3600))-(NumMinutos*60);

printf("\n\t%d segundos corresponden a %d horas",NumSegundos,NumHoras);
printf(" %d minutos y %d segundos\n\n",NumMinutos, NumSegundos2);

system("PAUSE");
return 0;
}

```

Salida

```

Ingrese un cierto número de segundos :12574
12574 segundos corresponden a 3 horas 29 minutos y 34 segundos
Presione una tecla para continuar . . .

```

Observaciones:

1) Recordemos que 1 hora = 3600 segundos y que 1 minuto = 60 segundos.

2) Si dividimos el total de segundos por 3600 obtendremos el total de horas asociadas, pero nos sobrará la diferencia entre el total de segundos y los segundos asociados con las horas calculadas. Esos segundos sobrantes debemos convertirlos a minutos, para ello se dividen por 60, porque 60 segundos es un minuto. Finalmente, la cantidad de segundos que sobran es el total de segundos ingresados menos los segundos asociados a las horas y los minutos ya calculados. □

31

Escriba un programa en C que calcule y muestre por pantalla la suma de los restos de las divisiones $5/4$ y $3/8$

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int resto1, resto2;

    resto1=5%4;
    resto2=3%8;

    printf("\nLa suma de los restos de 5/4 y 3/8 es %d",resto1+resto2);

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida

```
La suma de los restos de 5/4 y 3/8 es 4
Presione una tecla para continuar . . .
```

Observaciones:

1) El operador módulo % permite calcular el resto de la división entre dos números enteros. Por ejemplo, $7\%2 = 1$, pues el resto de dividir 7 por 2 es justamente 1.

2) El operador módulo entrega como resultado siempre un número entero, por tal motivo las variables resto1 y resto2 son declaradas como enteras.

33

Escriba un programa en C que calcule y muestre por pantalla el valor de la función $f(x) = \frac{\sqrt[5]{x} e^{x^3}}{x^2+1} + \sqrt{\pi}$ donde x es un valor entero ingresado por el usuario.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    int x; float ValorFuncion;
    printf("\nIngrese un valor entero x:");
    scanf("%i",&x);
    ValorFuncion=(pow(x,1.0/5)*exp(pow(x,3)))/(pow(x,2)+1)+pow(M_PI,1.0/2);
    printf("\nEl valor de la funci%cn en x=%i es %.2f\n\n",162,x,ValorFuncion);
    system("PAUSE");
    return 0;
}
```

Salida 1

```
Ingrese un valor entero x:1
El valor de la función en x=1 es 3.13
Presione una tecla para continuar . . .
```

Salida 2

```
Ingrese un valor entero x:5
El valor de la función en x=5 es 1.#J
Presione una tecla para continuar . . .
```



```

int main()
{
    int x;
    float Rad, Resultado;

    printf("\nIngrese x en grados : ");
    scanf("%i",&x);

    Rad=x*M_PI/180;

    Resultado=(4e4/7e-7)*pow(sin(Rad),3)/(1+tan(Rad));

    textbackground(WHITE);
    textcolor(LIGHTBLUE);
    printf("\n\nEl resultado es : %e\n\n",Resultado);

    printf("\n\n");
    system("PAUSE");
    return 0;
}

```

Salida

```

Ingrese x en grados : 45

El resultado es : 1.010153e+010

Presione una tecla para continuar . . .

```

Observaciones:

- 1) La función $\tan(x)$, presente en la biblioteca math.h, permite calcular la tangente del argumento real x .
- 2) Dado que x fue ingresado en grados, fue necesaria la conversión a radianes antes de calcular el resultado, pues recordemos que C trabaja en radianes.
- 3) Se usó el especificador de formato `%e` para expresar el resultado en notación científica.

35

Use C para calcular y mostrar por pantalla el resultado de

$\frac{4 \times 10^6}{3 \times 10^n} + \ln\left(\left|\sqrt[5]{n^2}\right|\right)$, usando notación científica, donde n es un número entero ingresado por el usuario.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    int n;
    float Resultado;
    printf("\nIngrese n : ");
    scanf("%i",&n);

    Resultado=(4e6/(3*pow(10,n))+log(fabs(pow(n,2./5))));

    printf("\nResultado = %E\n",Resultado);

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida

```
Ingrese n : 5
Resultado = 1.397711E+001
Presione una tecla para continuar . . .
```

Observaciones:

1) La notación científica en C permite sólo el uso de valores constantes, es decir, en el ejercicio anterior no se puede escribir

```
Resultado=(4e6/3en+log(fabs(pow(n,2./5)));
```

Por tal razón se incluyó la función pow.

2) Siempre es posible anidar funciones como en la expresión:

```
log(fabs(pow(n,2./5))) 
```

38

Escriba un programa en C que calcule y muestre por pantalla el valor $f(4)$ para

$$f(x) = \frac{1-x^2}{1+x^3}$$

Solución:

Código Fuente

```
##include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    const x=4;
    float fx;
    fx=(1-pow(x,2))/(1+pow(x,3));
    printf("\nEl valor de la funci%cn evaluada en x=4 es %0.5f\n",162,fx);
    system("PAUSE");
    return 0;
}
```

Salida

```
El valor de la función evaluada en x=4 es -0.23077
Presione una tecla para continuar . . .
```

Observaciones:

1) Dado que debemos evaluar la función en un valor constante igual a 4, es que se escribe la línea const x=4;

2) Otra forma de calcular lo solicitado es incluyendo el valor 4 en la expresión de la función directamente, es decir

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    float fx;
    fx=(1-pow(4,2))/(1+pow(4,3));
    printf("\nEl valor de la funci%cn evaluada en x=4 es %0.5f\n",162,fx);
    system("PAUSE");
    return 0;
}
```

cuya salida es idéntica a la anterior.

3) Una tercera forma, de hacer lo mismo, es incluir la evaluación de la función en 4 directamente en el printf, es decir

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    printf("\nEl valor de la funci%cn evaluada en x=4 es %0.5f\n",162,(1-pow(4,2))/(1+pow(4,3)));
    system("PAUSE");
    return 0;
}
```

cuya salida es nuevamente idéntica a la señalada.

40

Escriba un programa en C que calcule y muestre por pantalla, en color verde brillante, el producto $(a + b)(c + d)$, donde a, b, c y d son números enteros ingresados por el usuario.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <conio2.h>

int main()
{
    int a,b,c,d;
    printf("\nIngrese primer valor entero : ");
    scanf("%i",&a);
    printf("\nIngrese segundo valor entero : ");
    scanf("%i",&b);
    printf("\nIngrese tercer valor entero : ");
    scanf("%i",&c);
    printf("\nIngrese cuarto valor entero : ");
    scanf("%i",&d);

    textbackground(WHITE);
    textcolor(LIGHTGREEN);
    printf("\n\t(%i+%i)(%i+%i) = %i\n",a,b,c,d,(a+b)*(c+d));

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida

```
Ingrese primer valor entero : 2
Ingrese segundo valor entero : 5
Ingrese tercer valor entero : 3
Ingrese cuarto valor entero : 8
      <2+5><3+8> = 77
Presione una tecla para continuar . . .
```

Observación:

En general, es apropiado mostrar por pantalla las operaciones realizadas para llegar al resultado exhibido. Por este motivo, se incluye en la instrucción

```
printf("\n\t(%i+%i)(%i+%i) = %i\n",a,b,c,d,(a+b)*(c+d));
```

el detalle de las operaciones.

44

Use C para solicitar al usuario el ingreso de las coordenadas (x, y) de un punto del plano y luego convertir tales coordenadas cartesianas a polares.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    float x, y, r, alfa;
    printf("\nIngreso de las coordenadas del punto (x,y)\n");
    printf("\n x : ");
    scanf("%f",&x);
```

```

printf("\n y : ");
scanf("%f",&y);

r=sqrt(x*x+y*y);
alfa=atan2(y,x);

alfa=180/M_PI*alfa;

printf("\nLas coordenadas polares son (r,alfa) = (%0.2f,%0.1f%c)",r,alfa,248);

printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida

```

Ingreso de las coordenadas del punto <x,y>
x : -1
y : -1
Las coordenadas polares son <r,alfa> = <1.41,-135.0º>
Presione una tecla para continuar . . .

```

Observaciones:

1) Si (x, y) son las coordenadas de un punto en el plano cartesiano, entonces (r, α) son las coordenadas polares de tal punto, donde

$$r = \sqrt{x^2 + y^2}$$

$$\alpha = \text{Arctg2}\left(\frac{y}{x}\right),$$

donde

$$\text{Arctg2}\left(\frac{y}{x}\right) = \begin{cases} \text{Arctg}\left(\frac{y}{x}\right) & , \text{ si } x > 0 \\ \text{Arctg}\left(\frac{y}{x}\right) + \pi & , \text{ si } x < 0 \text{ y } y \geq 0 \\ \text{Arctg}\left(\frac{y}{x}\right) - \pi & , \text{ si } x < 0 \text{ y } y < 0 \\ \frac{\pi}{2} & , \text{ si } x = 0 \text{ y } y > 0 \\ -\frac{\pi}{2} & , \text{ si } x = 0 \text{ y } y < 0 \\ \text{indefinido} & , \text{ si } x = 0 \text{ y } y = 0 \end{cases}$$

2) La función $\text{atan}(z)$ permite calcular el arcotangente de z en radianes en el intervalo $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$

La función $\text{atan2}(z,w)$ calcula el arcotangente de $\frac{z}{w}$ expresado en radianes en el intervalo $[-\pi, \pi]$. Esta función tiene en consideración el cuadrante en el que termina el ángulo (recordemos que el ángulo comienza a medirse siempre desde la parte positiva del eje horizontal, como negativo en sentido horario y positivo en sentido antihorario).

3) En este ejercicio se usó la misma variable (alfa) para almacenar el valor en radianes primero y luego en grados. Esto es posible en programación porque la línea de código `alfa=180/M_PI*alfa;` se lee como "el *valor actual* de alfa es igual a 180 dividido por π multiplicado por el *valor anterior* de alfa", lo que nos muestra que una misma variable puede adoptar distintos valores a medida que avanzamos en el código fuente. \square

45

Use C para calcular la suma, la resta, el producto y el cociente de dos números complejos ingresados por el usuario. Considere dos decimales.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    float a, b, c, d;

    printf("\nIngrese parte real primer n%cmero : ",163);
    scanf("%f",&a);

    printf("\nIngrese parte compleja primer n%cmero : ",163);
    scanf("%f",&b);

    printf("\nIngrese parte real segundo n%cmero : ",163);
    scanf("%f",&c);
```



```

printf("\nIngrese parte compleja segundo n°mero : ",163);
scanf("%f",&d);

printf("\n");
printf("\nLa suma es %0.2f + %0.2f i\n",a+c,b+d);
printf("\nLa resta es %0.2f + %0.2f i\n",a-c,b-d);
printf("\nEl producto es %0.2f + %0.2f i\n",a*c-b*d,a*d+b*c);
printf("\nEl cuociente es %0.2f + %0.2f i\n",(a*c+b*d)/(c*c+d*d),(b*c-a*d)/(c*c+d*d));

printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida

```

Ingrese parte real primer número : 3
Ingrese parte compleja primer número : 2
Ingrese parte real segundo número : -1
Ingrese parte compleja segundo número : 4

La suma es 2.00 + 6.00 i
La resta es 4.00 + -2.00 i
El producto es -11.00 + 10.00 i
El cuociente es 0.29 + -0.82 i

Presione una tecla para continuar . . .

```

Observaciones:

1) Sean $z_1 = a + bi$ y $z_2 = c + di$ dos números complejos, donde i es la unidad imaginaria. Se tiene que:

$$z_1 + z_2 = (a + c) + (b + d)i$$

$$z_1 - z_2 = (a - c) + (b - d)i$$

$$z_1 z_2 = (ac - bd) + (ad + bc)i$$

$$\frac{z_1}{z_2} = \frac{ac+bd}{c^2+d^2} + \frac{bc-ad}{c^2+d^2} i$$

2) En el programa se calculan la parte real y la parte imaginaria de cada número complejo solicitado, y estos resultados se muestran usando un formato apropiado.

46

Escriba un programa en C que muestre la fecha y hora actual por pantalla.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    time_t momento;

    momento=time(NULL);

    printf("\nLa fecha y hora actual es %s\n",asctime(localtime(&momento)));

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida

```
La fecha y hora actual es Fri Jun 26 11:58:59 2015

Presione una tecla para continuar . . .
```

Observaciones:

1) El tipo de dato `time_t` sirve para representar fechas y horas, y en el caso del ejercicio se declara la variable `momento` de tipo `time_t`

2) La función `time()` devuelve la fecha/hora actual o -1 en caso de no ser posible.

3) La función `asctime` convierte una estructura horaria en una cadena de caracteres para, en este ejercicio, ser exhibida por pantalla, usando el especificador de formato `%s`

4) En este ejercicio la función `localtime` convierte la variable momento a la fecha y hora actual.

48

Escriba un programa en C que muestre por pantalla los valores de la función

$$f(x) = \begin{cases} x^3 - 3 & , \text{ si } x < -4 \\ \sqrt{x + 4} & , \text{ si } -4 \leq x < 4 \\ \cos\left(\frac{1}{x+4}\pi\right) & , \text{ si } x \geq 4 \end{cases}$$

en $x = -6$, en $x = 1$ y en $x = 2\pi$.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    float x;
    x=-6;
    printf("\n f(%0.1f) = %f\n",x,pow(x,3)-3);

    x=1;
    printf("\n f(%0.1f) = %f\n",x,sqrt(x+4));

    x=2*M_PI;
    printf("\n f(%0.1f) = %f\n",x,cos(1/(x+4)*M_PI));
```

```
printf("\n\n");
system("PAUSE");
return 0;
}
```

Salida

```
f<-6.0> = -219.000000
```

```
f<1.0> = 2.236068
```

```
f<6.3> = 0.953694
```

```
Presione una tecla para continuar . . .
```

Observaciones:

1) En este ejercicio el programador es quien decide cuál tramo de la función es el elegido para cada uno de los valores de x .

2) El valor $x = 2\pi$ es aproximadamente igual a 6.28 que es superior a 4, por lo que corresponde evaluar en el tercer o último tramo ($x \geq 4$).

GUÍA DE PRÁCTICO N° 6

ESTRUCTURAS CONDICIONALES

Teoría Estructuras Condicionales

T 23 Instrucciones de Control

C es un lenguaje de programación que permite trabajar de manera estructurada, gracias a las instrucciones de control con que cuenta, que permiten controlar el flujo del programa, es decir, el orden de ejecución de las instrucciones de un programa.

Las estructuras de control que se pueden utilizar en C son:

- a) Estructuras secuenciales. Por defecto las instrucciones de un programa se ejecutan de manera secuencial.
- b) Estructuras condicionales o selectivas. Si se cumplen o no ciertas condiciones, el programa ejecutará un conjunto u otro de instrucciones.
- c) Estructuras iterativas o repetitivas. Permiten ejecutar un conjunto de instrucciones un cierto número de veces.

T 24 Estructura Condicional **if (condicion) ... else ...**

La estructura condicional **if(condicion) ... else ...**, que se lee **si(condicion) ... en caso contrario ...**, permite ejecutar ciertas instrucciones dependiendo de si **condicion** es verdadera o falsa.

Las sintaxis, o formas correctas de escribir las instrucciones, relacionadas con **if** son las siguientes

Sintaxis 1:

```
if(condicion) sentencia;
```

En este caso, si condicion es verdadera, entonces se ejecuta sentencia.

Sintaxis 2:

```
if(condicion) { bloque de sentencias }
```

En este caso, si condicion es verdadera se ejecuta el bloque de sentencias.

Sintaxis 3:

```
if(condicion) sentencia1;  
else sentencia2;
```

En este caso, si condicion es verdadera se ejecuta sentencia1, en caso contrario se ejecuta sentencia2.

Sintaxis 4:

```
if(condicion) { bloque de sentencias1 }  
else { bloque de sentencias2 }
```

En este caso, si condicion es verdadera se ejecuta el bloque de sentencias1, en caso contrario se ejecuta el bloque de sentencias2.

25 **Operador Condicional**

El lenguaje C permite el uso del llamado operador condicional, que es una forma abreviada de escribir una estructura condicional if

La sintaxis del operador condicional es

```
condicion? expresion1: expresion2;
```

En este caso, si condicion es verdadera, entonces se evalúa la expresion1 y se convierte en el resultado, en caso contrario se evalúa la expresion2 y esta se convierte en el resultado.

T 26 **Estructura Condicional switch**

La estructura condicional switch (interruptor) permite seleccionar las acciones a realizar de acuerdo al valor que tome una variable.

Es usada con mucha frecuencia para permitirle al usuario elegir una entre varias opciones disponibles.

La sintaxis de la estructura condicional switch es

```
switch(variable)
{
case valor_1: {acciones_1;
                  break;}
case valor_2: {acciones_2;
                  break;}
                  .
                  .
                  .
case valor_n: {acciones_n;
                  break;}
default: {acciones_default;
                  break;}
}
```

En este caso, si variable toma el valor_1, se ejecutan las acciones_1, si toma el valor_2, se ejecutan las acciones_2, y así sucesivamente hasta el valor_n.

Si el valor que toma la variable no coincide con ninguno de los valores mencionados en los casos (case), entonces se ejecutan acciones_default.

Ejercicios

Estructuras Condicionales

[1] Escriba un programa en C que muestre por pantalla un mensaje indicando si el número entero positivo ingresado por el usuario es par o impar.

[2] Escriba un programa en C que solicite al usuario el ingreso de un número entero y luego muestre por pantalla si el número ingresado es múltiplo de 6 y 9.

[3] Escriba un programa en C que calcule y muestre por pantalla el promedio de tres notas ingresadas por el usuario. Además se debe indicar si el alumno aprobó o no la signatura, sabiendo que si el promedio es superior o igual a 4 se está aprobado.

[4] Escriba un programa en C que permita evaluar la función

$$f(x) = \begin{cases} x^3 - 3 & , \text{ si } x < -4 \\ \sqrt{x + 4} & , \text{ si } -4 \leq x < 4 \\ \cos\left(\frac{1}{x+4}\pi\right) & , \text{ si } x \geq 4 \end{cases}$$

en un x ingresado por el usuario.

[5] Escriba un programa en C que solicite al usuario el ingreso de las coordenadas de un punto del plano y muestre por pantalla un mensaje que indique si tal punto está fuera, dentro o en la elipse $\frac{x^2}{6} + 3y^2 = 4$

[6] Escriba un programa en C que solicite al usuario el ingreso de los pesos, en kilogramos, de 5 ocupantes de un vehículo y emita un mensaje indicando el peso total y si tal peso total excede un cuarto de tonelada.

[7] Escriba un programa en C que calcule y muestre por pantalla, con 3 decimales, el valor de la función

$$f(r) = \begin{cases} \frac{1}{\sqrt{\sqrt{2}-r}} & , \text{ si } r < \sqrt{2} \\ r^4 - \frac{160}{3} & , \text{ si } \sqrt{2} \leq r \leq 7 \\ \ln(r-3) & , \text{ si } r > 7 \end{cases}$$

para un valor de r real ingresado por el usuario.

[8] Escriba un programa en C que solicite al usuario el ingreso de su edad y si es un hombre ingrese un -1 y si es una mujer ingrese un -2. Además muestre por pantalla cuántos años le faltan para jubilar. Suponga que un hombre jubila a los 65 años y una mujer a los 60 años. Si el usuario ya está jubilado el programa debe indicarlo.

[9] Use C para mostrar por pantalla si un número entero ingresado por el usuario es divisible por 7, y si lo es multiplicarlo por 3

[10] Escriba un programa en C que solicite al usuario el ingreso de la longitud de dos lados de un triángulo y del ángulo entre ellos expresado en grados, y calcule el tercer lado usando el teorema del coseno. El programa debe mostrar un mensaje en verde que indique si el triángulo es rectángulo isósceles. Recuerde que C trabaja en radianes.

[11] Escriba un programa en C que solicite al usuario el ingreso de un ángulo en grados y, calcule y muestre por pantalla el valor de la función

$$f(\alpha) = \begin{cases} \cos(\alpha) + 1 & , \text{ si } \alpha \leq 0 \\ \sin(\alpha) - 1 & , \text{ si } 0 < \alpha \leq 90 \\ \cos(2\alpha) - \sin(2\alpha) & , \text{ si } \alpha > 90 \end{cases}$$

[12] Escriba un programa en C que calcule y muestre por pantalla el área y el perímetro de un rectángulo cuyas longitudes de los lados son ingresadas por el usuario. El programa debe mostrar además un mensaje por pantalla en color rojo que indique cuando el rectángulo sea un cuadrado.

[13] Escriba un programa en C que solicite al usuario el ingreso del número de años trabajados y su sueldo al empezar a trabajar, y determine el sueldo actual si se sabe que ha aumentado un 3.8% cada año con respecto al año anterior.
Indique si el sueldo actual superó el millón de pesos.

[14] Escriba un programa en C que solicite al usuario el ingreso del número de años trabajados y determine el número de quinquenios trabajados y cuántos años le faltan para completar un nuevo quinquenio.
Si por el primer quinquenio se paga una cantidad base de 155600 pesos, reajutable en un 7% al finalizar éste, calcule y muestre por pantalla la cantidad, en pesos, acumulada a través de los años de trabajo por concepto de quinquenios, si los restantes quinquenios se reajustan también en un 7% con respecto al anterior.

[15] Escriba un programa en C que solicite al usuario el ingreso de un número entero del 1 al 12 y muestre por pantalla el mes asociado a ese número y el número de días asociado al mes elegido. Por ejemplo, si el usuario ingresa 3 el programa debe mostrar el mensaje "MARZO tiene 31 días".
Si el usuario elige el número 2, entonces el programa debe mostrar el mensaje: "FEBRERO tiene 28 ó 29 días".

[16] Use el lenguaje C para solicitarle al usuario el ingreso de dos números reales y calcular la suma, la resta, el producto y el cociente de tales números. El programa debe emitir un mensaje apropiado cuando el denominador del cociente sea cero.

[17] Escriba un programa que determine el menor de dos números enteros ingresados por el usuario.

[18] Escriba un programa en C que solicite al usuario el ingreso de tres números enteros y muestre por pantalla los mismos números, pero ordenados de menor a mayor.

[19] Use C para determinar si dos números enteros ingresados por el usuario están en orden creciente o decreciente.

[20] Escriba un programa en C que solicite al usuario el ingreso de las coordenadas de un punto del plano y luego muestre un mensaje por pantalla que indique el cuadrante al cual pertenece el punto o si tal punto está en alguno de los ejes cartesianos.

Además debe calcular y mostrar por pantalla la distancia del punto ingresado al origen, e indicar si tal distancia es menor que $\sqrt{3}$

[21] Escriba un programa en C que calcule y muestre por pantalla, con 3 decimales, el valor de la función

$$f(x) = \begin{cases} \frac{1-x}{\sqrt{x^2-1}} & , \text{ si } x > 1 \\ x + 1 & , \text{ si } 0 \leq x \leq 1 \\ \ln(|x|) & , \text{ si } x < 0 \end{cases}$$

para un valor de x real ingresado por el usuario.

[22] Escriba un programa en C que calcule y muestre por pantalla el valor de

$$f(x) = \begin{cases} 1 & , \text{ si } x \geq 0 \\ -1 & , \text{ si } x < 0 \end{cases}$$

para un valor de x real ingresado por el usuario.

[23] Escriba un programa en C que calcule y muestre por pantalla el valor de

$$f(x) = \begin{cases} 1 & , \text{ si } -2 \leq x < 3 \\ -1 & , \text{ en otro caso} \end{cases}$$

para un valor de x real ingresado por el usuario.

[24] Escriba un programa en C que calcule y muestre por pantalla el valor de

$$f(x) = \begin{cases} 1 & , \text{ si } -2 \leq x < 3 \\ 0 & , \text{ si } 5 < x \leq 10 \\ -1 & , \text{ en otro caso} \end{cases}$$

para un valor de x real ingresado por el usuario.

[25] Escriba un programa en C que solicite al usuario el ingreso de un número del 1 al 7 y muestre por pantalla el día de la semana asociado a ese número, donde 1 corresponde a Domingo, 2 a Lunes, y así sucesivamente.

[26] Escriba un programa en C que calcule y muestre por pantalla el valor de la función $f(x) = \frac{\pi - x^3}{\sqrt{x^2 - 1}}$ para un x real ingresado por el usuario. Incorpore restricciones apropiadas al usuario.

[27] Escriba una aplicación en C que lea tres números enteros. Si el primero es negativo, entonces muestre por pantalla el producto de los tres números. En caso contrario, muestre la suma de los números.

[28] Escriba un programa en C que solicite al usuario el ingreso de su edad y dependiendo de ésta muestra un mensaje apropiado en color, como se indica en la siguiente tabla:

Edad (años)	Mensaje	Color
0 a 1	Eres un bebé	Verde
2 a 15	Eres un niño	Azul
16 a 20	Eres un adolescente	Café
21 a 59	Eres un adulto	Rojo
60 o más	Eres de la tercera edad	Amarillo

[29] Escriba un programa en C que muestre por pantalla, con 3 decimales, el valor de la función $f(t)$ con

$$f(t) = \begin{cases} 10^2 \frac{t}{1+t} & , \text{ si } t > 2 \times 10^4 \\ \frac{e^{10}}{t} & , \text{ si } 10 < t \leq 2 \times 10^4 \\ t^2 + \frac{t^3-1}{t^2} & , \text{ si } t \leq 10 \end{cases}$$

donde t es un valor real ingresado por el usuario.

[30] Escriba un programa en C que determine si un año ingresado por el usuario es bisiesto o no.

(Indicación: Un año es bisiesto si es divisible por 4, pero no si es divisible por 100, a menos que también sea divisible por 400)

[31] Escriba un programa que solicite al usuario el ingreso de la longitud de dos lados de un triángulo y del ángulo entre ellos expresado en grados, y calcule y muestre por pantalla la longitud del tercer lado y los otros dos ángulos, en grados, usando el teorema del coseno. El programa debe mostrar un mensaje en verde que indique si el triángulo es rectángulo isósceles. Note que C trabaja en radianes.

[32] Escriba un programa en C que pregunte el año actual y la edad de una persona y calcule la edad de esa persona en el año 2047. Además se debe mostrar un mensaje por pantalla que indique si tal persona superará los 50 años en el 2047. Asuma que la edad que se ingresa es la edad que se cumple en el año actual.

[33] Escriba un programa en C que, haciendo uso de la orden SWITCH, muestre un menú que permita al usuario elegir entre las opciones siguientes: calcular el área y el perímetro de un círculo, rectángulo o triángulo. El usuario es quien ingresa el radio del círculo, el ancho y el alto del rectángulo, y los lados del triángulo. Muestre por pantalla los resultados.

[34] Prepara un programa en C que permita al usuario elegir una de las tres opciones siguientes, ingresando el usuario los datos que sean necesarios:

a) Dibujar un rectángulo de 3 caracteres de ancho por 10 de largo, usando asteriscos de color rojo.

b) Calcular y mostrar por pantalla $x \cdot y$, con x entero tal que $0 < x < 100$, y y real tal que $-10 < y < 5000$.

c) Calcular y mostrar por pantalla la suma de los primeros n términos de la progresión aritmética :

1, 4, 7, 10,

[35] Prepara un programa en C que permita al usuario elegir una de las dos opciones siguientes, ingresando el usuario los datos que sean necesarios:

a) Calcular el área $A = \sqrt{s(s-a)(s-b)(s-c)}$ de un triángulo cuyos lados son a , b y c , y donde $s = \frac{a+b+c}{2}$. Verifique que $A \geq 0$.

b) Calcular $\frac{x^2}{x+y}$, con x entero tal que $|x| < 10$, y y real tal que $|y| > 2$.

[36] Escriba un programa en C que calcule y muestre por pantalla el promedio, la desviación estándar y el rango de 3 notas, del 1 al 7, ingresadas por el usuario.

[37] Escriba un programa en C que lea un punto del plano (x, y) y determine si tal punto pertenece a la curva $\frac{(x-1)}{4} - \frac{(y+2)}{3} = 1$

[38] Escriba un programa en C que permita al usuario elegir una de entre 26 diferentes conversiones y muestre, a todo color, los resultados correspondientes

[39] Escriba un programa en C que muestre por pantalla las raíces de la ecuación $ax^2 + bx + c = 0$, donde a , b y c son números reales ingresados por el usuario. Considere todos los casos posibles.

PROBLEMAS RESUELTOS ESTRUCTURAS CONDICIONALES

1

Escriba un programa en C que muestre por pantalla un mensaje indicando si el número entero positivo ingresado por el usuario es par o impar.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int n;
    printf("\nIngrese un valor entero n : ");
    scanf("%d",&n);

    if(n%2==0) printf("\nEl número %i es par\n",n);
    else printf("\nEl número %i es impar\n",n);

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```


Salida 1

```
Ingrese un valor entero n : 6
El número 6 es par
Presione una tecla para continuar . . .
```

Salida 2

```
Ingrese un valor entero n : 7
El número 7 es impar
Presione una tecla para continuar . . .
```

Salida 3

```
Ingrese un valor entero n : -3
El número -3 es impar
Presione una tecla para continuar . . .
```

Observaciones:

- 1) IF significa SI y ELSE significa EN CASO CONTRARIO
- 2) Un número entero es par cuando al dividirlo por 2 el resto de la división es 0, e impar si el resto es distinto de cero.
- 3) Una de las estructuras condicionales en C es `if (condicion) ... else ...` que nos permite tomar ciertas decisiones de acuerdo a si la condición incluida es verdadera o falsa.
En el ejercicio la condición es `n%2==0` que es verdadera cuando el resto de dividir `n` por 2 es cero, es decir, si la condición es verdadera se muestra por pantalla el mensaje que indica que `n` es par, y en caso contrario (`else`) se muestra el mensaje que indica que `n` es impar.
- 4) No confundir `==` que es una igualdad lógica usada en condiciones, con `=` que es usada en las asignaciones.

2

Escriba un programa en C que solicite al usuario el ingreso de un número entero y luego muestre por pantalla si el número ingresado es múltiplo de 6 y 9.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int m;
    printf("\nIngrese un número entero : ");
    scanf("%i",&m);

    if(m%6==0 && m%9==0)
        printf("\nEl número %i es múltiplo de 6 y 9\n",m);

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida

```
Ingrese un número entero : 18
El número 18 es múltiplo de 6 y 9
Presione una tecla para continuar . . .
```

Observaciones:

1) En C existen operadores o conectores lógicos que permiten escribir una proposición compleja o molecular. En el ejercicio la condición del if en realidad

son dos condiciones obligatorias, por tal motivo se usa el operador lógico y, que se escribe en C como &&

2) Recordemos que los operadores lógicos en C son:

Op. Lógico	En C
y	&&
o	
no	not

3) Si en el programa ingresamos un valor que no es múltiplo de 6 ni de 9, entonces la salida exhibe nada, como se muestra a continuación:

```
Ingrese un número entero : 15
Presione una tecla para continuar . . .
```

Esto muestra la necesidad de emitir un mensaje que indique tal situación, como se hace en el siguiente programa

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int m;
    printf("\nIngrese un número entero : ",163);
    scanf("%i",&m);

    if(m%6==0 && m%9==0)
        printf("\nEl número %i es múltiplo de 6 y 9\n",163,m,163);
    else
        printf("\nEl número %i no es múltiplo de 6 %c de 9\n",163,m,163,162);

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

cuya salida es

```
Ingrese un número entero : 15
El número 15 no es múltiplo de 6 ó de 9
Presione una tecla para continuar . . .
```

4) La aparición o no del else en una estructura condicional if es opcional y depende de las condiciones del problema a resolver.

5) Es muy común cometer el error de escribir punto y coma (;) al final de la condición en un if, es decir, es común que se anote, por ejemplo:

```
if(m%6==0 && m%9==0) ;
printf("\nEl número %i es múltiplo de 6 y 9\n",m,163);
```

Lo anterior es erróneo, porque el mensaje que aparece en el printf se exhibirá por pantalla de todas formas, y sin considerar la condición de que el número sea múltiplo de 6 y de 9.

3

Escriba un programa en C que calcule y muestre por pantalla el promedio de tres notas ingresadas por el usuario. Además se debe indicar si el alumno aprobó o no la signatura, sabiendo que si el promedio es superior o igual a 4 se está aprobado.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    float nota1, nota2, nota3, promedio;
```

```

printf("\nNota 1 : ");
scanf("%f",&nota1);
printf("\nNota 2 : ");
scanf("%f",&nota2);
printf("\nNota 3 : ");
scanf("%f",&nota3);

promedio=(nota1+nota2+nota3)/3;

if(promedio>=4.0)
    printf("\nEl alumno est%c aprobado\n",160);
else
    printf("\nEl alumno est%c reprobado\n",160);

printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida 1

```

Nota 1 : 3.5
Nota 2 : 4.7
Nota 3 : 4.2
El alumno está aprobado
Presione una tecla para continuar . . .

```

Salida 2

```

Nota 1 : 2.5
Nota 2 : 3.8
Nota 3 : 4.3
El alumno está reprobado
Presione una tecla para continuar . . .

```

Observaciones:

1) La estructura condicional a usar en el programa se puede escribir del siguiente modo:

```
if(promedio>=4.0)
    printf("\nEl alumno est%c aprobado\n",160);
else
    printf("\nEl alumno est%c reprobado\n",160);
```

O de este otro modo:

```
if(promedio>=4.0) printf("\nEl alumno est%c aprobado\n",160);
else printf("\nEl alumno est%c reprobado\n",160);
```

2) Notemos que al calcular el promedio, el numerador es un número real y el denominador un número entero, por lo que el cociente es real.

```
promedio=(nota1+nota2+nota3)/3; 
```

5

Escriba un programa en C que solicite al usuario el ingreso de las coordenadas de un punto del plano y muestre por pantalla un mensaje que indique si tal punto está fuera, dentro o en la elipse $\frac{x^2}{6} + 3y^2 = 4$

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main()
{
    float x, y;
```

```
    printf("\nIngreso de las coordenadas de un punto del plano\n");
```

```

printf("\n x : ");
scanf("%f",&x);
printf("\n y : ");
scanf("%f",&y);

if(x*x/6+3*y*y==4)
    printf("\nEl punto pertenece a la elipse\n");
else if(x*x/6+3*y*y>4)
    printf("\nEl punto est%c fuera de la elipse\n",160);
else
    printf("\nEl punto est%c dentro de la elipse\n",160);

printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida 1

```

Ingreso de las coordenadas de un punto del plano
x : 4
y : 6
El punto está fuera de la elipse
Presione una tecla para continuar . . .

```

Salida 2

```

Ingreso de las coordenadas de un punto del plano
x : 1
y : 1
El punto está dentro de la elipse
Presione una tecla para continuar . . .

```

Observaciones:

1) En el ejercicio aparecen if anidados, es decir, un if dentro de otro if. Esto es perfectamente válido y podemos tener tantos if anidados como queramos.

La estructura

```
if(x*x/6+3*y*y==4)
    printf("\nEl punto pertenece a la elipse\n");
else if(x*x/6+3*y*y>4)
    printf("\nEl punto est%c fuera de la elipse\n",160);
else
    printf("\nEl punto est%c dentro de la elipse\n",160);
```

es un ejemplo de un if (el primero) cuyo else asociado (el primero) cuenta con otro if (el segundo) que a su vez tiene un else asociado (el segundo).

2) El trozo de código anterior también se podría haber escrito de la siguiente manera:

```
if(x*x/6+3*y*y==4)
    printf("\nEl punto pertenece a la elipse\n");
else
    if(x*x/6+3*y*y>4)
        printf("\nEl punto est%c fuera de la elipse\n",160);
    else
        printf("\nEl punto est%c dentro de la elipse\n",160);
```

que permite visualizar de mejor manera, gracias a la indentación o sangrado, la relación entre los if y los else correspondientes.

7

Escriba un programa en C que calcule y muestre por pantalla, con 3 decimales, el valor de la función

$$f(r) = \begin{cases} \frac{1}{\sqrt{\sqrt{2}-r}} & , \text{ si } r < \sqrt{2} \\ r^4 - \frac{160}{3} & , \text{ si } \sqrt{2} \leq r \leq 7 \\ \ln(r-3) & , \text{ si } r > 7 \end{cases}$$

para un valor de r real ingresado por el usuario.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    float r, fr;

    printf("\n r = "); scanf("%f",&r);

    if(r<sqrt(2))
    {
        fr=1/sqrt(sqrt(2)-r);
        printf("\nf(%0.3f) = %0.3f\n",r,fr);
    }
    else if(r>=sqrt(2) && r<=7.0)
    {
        fr=pow(r,4)-160./3;
        printf("\nf(%0.3f) = %0.3f\n",r,fr);
    }
    else
    {
        fr=log(r-3);
        printf("\nf(%0.3f) = %0.3f\n",r,fr);
    }

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida 1

```
r = 1
f<1.000> = 1.554
Presione una tecla para continuar . . .
```

Salida 2

```
r = 3.6
f<3.600> = 114.628
Presione una tecla para continuar . . .
```

Salida 3

```
r = 10
f<10.000> = 1.946
Presione una tecla para continuar . . .
```

Observaciones:

1) En C todas las sentencias terminan con punto y coma, y si existen dos o más sentencias asociadas a un if, entonces es obligatorio el uso de paréntesis de llave que encierren el bloque de código relacionado con el if, como aparece, por ejemplo, en

```
if(r<sqrt(2))
{
    fr=1/sqrt(sqrt(2)-r);
    printf("\nf(%0.3f) = %0.3f\n",r,fr);
}
```

2) Siempre es posible escribir más de una sentencia en una sola línea, por ejemplo,

```
printf("\n r = "); scanf("%f",&r);
```

3) El programa anterior se podría haber escrito de la siguiente manera:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    float r, fr;

    printf("\n r = "); scanf("%f",&r);

    if(r<sqrt(2))
        fr=1/sqrt(sqrt(2)-r);
    else if(r>=sqrt(2) && r<=7.0)
        fr=pow(r,4)-160./3;
    else
        fr=log(r-3);

    printf("\nf(%0.3f) = %0.3f\n",r,fr);
    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Este programa, a diferencia de la solución dada inicialmente, muestra el resultado de la evaluación de la función al final del programa, lo que es perfectamente factible dado que la función es evaluada sólo una vez dependiendo del valor de r ingresado por el usuario.

8

Escriba un programa en C que solicite al usuario el ingreso de su edad y si es un hombre ingrese un -1 y si es una mujer ingrese un -2. Además muestre por pantalla cuántos años le faltan para jubilar. Suponga que un hombre jubila a los 65 años y una mujer a los 60 años. Si el usuario ya está jubilado el programa debe indicarlo.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int edad, hm, faltan;
    printf("\nIngrese su edad en años : ",164);
    scanf("%i",&edad);

    printf("\nIndique si es hombre (-1) o mujer (-2) : ");
    scanf("%i",&hm);

    if(hm===-1)
    {
        faltan=65-edad;
        if(faltan<=0) printf("\nUsted ya está jubilado\n",160);
        else printf("\nLe faltan %i años para jubilarse\n",faltan,164);
    }
    if(hm===-2)
    {
        faltan=60-edad;
        if(faltan<=0) printf("\nUsted ya está jubilada\n",160);
        else printf("\nLe faltan %i años para jubilarse\n",faltan,164);
    }

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida 1

```
Ingrese su edad en años : 48
Indique si es hombre (-1) o mujer (-2) : -1
Le faltan 17 años para jubilar
Presione una tecla para continuar . . .
```

Salida 2

```
Ingrese su edad en años : 55
Indique si es hombre (-1) o mujer (-2) : -2
Le faltan 5 años para jubilar
Presione una tecla para continuar . . .
```

Salida 3

```
Ingrese su edad en años : 78
Indique si es hombre (-1) o mujer (-2) : -1
Usted ya está jubilado
Presione una tecla para continuar . . .
```

Salida 4

```
Ingrese su edad en años : 63
Indique si es hombre (-1) o mujer (-2) : -2
Usted ya está jubilada
Presione una tecla para continuar . . .
```

Observaciones:

1) En este ejercicio se usaron dos if separados para decidir si el usuario es hombre o mujer, porque de ese modo nos aseguramos que el usuario ingrese -1

ó -2 y no otros valores, además para tales valores y no otros realizamos las acciones correspondientes

2) La variable faltan permite saber cuántos años faltan para jubilar y además saber si ya se está jubilado (faltan<=0)

11

Escriba un programa en C que solicite al usuario el ingreso de un ángulo en grados y, calcule y muestre por pantalla el valor de la función

$$f(\alpha) = \begin{cases} \cos(\alpha) + 1 & , \text{ si } \alpha \leq 0 \\ \sin(\alpha) - 1 & , \text{ si } 0 < \alpha \leq 90 \\ \cos(2\alpha) - \sin(2\alpha) & , \text{ si } \alpha > 90 \end{cases}$$

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    float angulo, falfa;
    printf("\nIngrese un %cngulo en grados : ",160);
    scanf("%f",&angulo);

    if(angulo<=0.0) falfa=cos(M_PI/180*angulo)+1;
    else if(angulo>0 && angulo<=90) falfa=sin(M_PI/180*angulo)-1;
    else falfa=cos(2*M_PI/180*angulo)-sin(2*M_PI/180*angulo);

    printf("\n f(%0.1f%c) = %0.3f\n",angulo,248,falfa);
```

```
printf("\n\n");
system("PAUSE");
return 0;
}
```

Salida 1

```
Ingrese un ángulo en grados : 45
f<45.0°> = -0.293
Presione una tecla para continuar . . .
```

Salida 2

```
Ingrese un ángulo en grados : -30
f<-30.0°> = 1.866
Presione una tecla para continuar . . .
```

Salida 3

```
Ingrese un ángulo en grados : 180
f<180.0°> = 1.000
Presione una tecla para continuar . . .
```

Observaciones:

- 1) Los identificadores no pueden incluir tildes, por ejemplo, angulo es un nombre de variable lícito, pero ángulo ya no lo es.
- 2) Los ángulos deben ser convertidos a radianes al momento de evaluar la función
- 3) El código ASCII 248 permite la inclusión del símbolo de grado. ◻

12

Escriba un programa en C que calcule y muestre por pantalla el área y el perímetro de un rectángulo cuyas longitudes de los lados son ingresadas por el usuario. El programa debe mostrar además un mensaje por pantalla en color rojo que indique cuando el rectángulo sea un cuadrado.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <conio2.h>

int main()
{
    float a,b, area, perimetro;

    printf("\nIngrese las longitudes de los lados del rectángulo\n",160);
    printf("\n a : ");
    scanf("%f",&a);
    printf("\n b : ");
    scanf("%f",&b);

    area=a*b;
    perimetro=2*(a+b);

    printf("\nEl área es %0.3f\n",160,area);
    printf("\nEl perímetro es %0.3f\n",161,perimetro);

    if(a==b)
    {
        textbackground(WHITE);
        textcolor(RED);
        printf("\nEs un cuadrado\n");
    }
}
```



```
printf("\n\n");
system("PAUSE");
return 0;
}
```

Salida 1

```
Ingrese las longitudes de los lados del rectángulo
a : 3
b : 7
El área es 21.000
El perímetro es 20.000
Presione una tecla para continuar . . .
```

Salida 2

```
Ingrese las longitudes de los lados del rectángulo
a : 4
b : 4
El área es 16.000
El perímetro es 16.000
Es un cuadrado
Presione una tecla para continuar . . .
```

Observaciones:

1) Si un rectángulo tiene lados cuyas longitudes son a y b , entonces su área es $a \cdot b$ y su perímetro es $2 \cdot (a + b) = 2a + 2b$

2) La variable `area` almacena el área del rectángulo y la variable `perimetro` almacena el perímetro.

13

Escriba un programa en C que solicite al usuario el ingreso del número de años trabajados y su sueldo al empezar a trabajar, y determine el sueldo actual si se sabe que ha aumentado un 3.8% cada año con respecto al año anterior. Indique si el sueldo actual superó el millón de pesos.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    int numant;
    int sueldoin;
    float sueldof;
    const float i=3.8/100;

    printf("\nIndique el número de años trabajados : ",163,164);
    scanf("%i",&numant);

    printf("\nIndique su sueldo inicial : ");
    scanf("%i",&sueldoin);

    sueldof=sueldoin*pow(1+i,numant);
    printf("\nEl sueldo final después de %i años trabajados es
    %0.1f",130,numant,164,sueldof);

    if(sueldof>1000000.0)
        printf("\nEl sueldo final superó el millón de pesos\n",162,162);

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida 1

```
Indique el número de años trabajados : 3
Indique su sueldo inicial : 157560
El sueldo final después de 3 años trabajados es 176213.0
Presione una tecla para continuar . . .
```

Salida 2

```
Indique el número de años trabajados : 29
Indique su sueldo inicial : 578900
El sueldo final después de 29 años trabajados es 1707366.6
El sueldo final superó el millón de pesos
Presione una tecla para continuar . . .
```

Observaciones:

1) El reajuste se declara como constante porque no cambia durante la ejecución del programa.

Además es una buena práctica de programación el asignar un identificador apropiado a las constantes, porque en el futuro pueden cambiar, por ejemplo, en este ejercicio el reajuste puede ser de un 5% o 2% y lo único que tendríamos que cambiar en el programa es la sentencia

```
const float i=3.8/100;
```

y no cada aparición de la constante 0.38

2) Si se aplica un reajuste del $i\%$ con respecto al sueldo del año anterior, entonces después de n años el sueldo final será igual a

$$\text{SueldoInicial} \cdot (1 + i)^n \quad \square$$

Escriba un programa en C que solicite al usuario el ingreso del número de años trabajados y determine el número de quinquenios trabajados y cuántos años le faltan para completar un nuevo quinquenio.

Si por el primer quinquenio se paga una cantidad base de 155600 pesos, reajutable en un 7% al finalizar éste, calcule y muestre por pantalla la cantidad, en pesos, acumulada a través de los años de trabajo por concepto de quinquenios, si los restantes quinquenios se reajustan también en un 7% con respecto al anterior.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int NumAT, NumQuin, NumAFaltan;
    float i=7./100;
    int CantInicial=155600;
    float CantAcum;
    printf("\nIngrese el número de años trabajados : ",163,164);
    scanf("%i",&NumAT);

    NumQuin = NumAT/5;
    NumAFaltan=5-NumAT%5;
    CantAcum=CantInicial/i*(pow(1+i,NumQuin)-1);

    printf("\nEl número de quinquenios trabajados es %i\n",163,NumQuin);

    if(NumAFaltan==1)
        printf("\nFalta %i año para completar un nuevo quinquenio\n",NumAFaltan,164);
    else
        printf("\nFaltan %i años para completar un nuevo quinquenio\n",NumAFaltan,164);

    printf("\nLa cantidad acumulada por concepto de quinquenios es %0.1f\n",CantAcum);
```

```
printf("\n\n");
system("PAUSE");
return 0;
}
```

Salida 1

```
Ingrese el número de años trabajados : 1
El número de quinquenios trabajados es 0
Faltan 4 años para completar un nuevo quinquenio
La cantidad acumulada por concepto de quinquenios es 0.0
Presione una tecla para continuar . . .
```

Salida 2

```
Ingrese el número de años trabajados : 11
El número de quinquenios trabajados es 2
Faltan 4 años para completar un nuevo quinquenio
La cantidad acumulada por concepto de quinquenios es 322092.0
Presione una tecla para continuar . . .
```

Salida 3

```
Ingrese el número de años trabajados : 9
El número de quinquenios trabajados es 1
Falta 1 año para completar un nuevo quinquenio
La cantidad acumulada por concepto de quinquenios es 155600.0
Presione una tecla para continuar . . .
```

Salida 4

```
Ingrese el número de años trabajados : 5
El número de quinquenios trabajados es 1
Faltan 5 años para completar un nuevo quinquenio
La cantidad acumulada por concepto de quinquenios es 155600.0
Presione una tecla para continuar . . .
```

Observaciones:

1) La sentencia $\text{NumQuin} = \text{NumAT}/5$; nos permite calcular el número de quinquenios porque es una división entera, es decir, nos entrega como resultado sólo el cociente sin decimales, por ejemplo, $6/4 = 1$ porque al dividir 6 por 4 obtenemos un entero.

2) La sentencia $\text{NumAFaltan} = 5 - \text{NumAT}\%5$; nos permite calcular el número de años que faltan para un nuevo quinquenio porque $\text{NumAT}\%5$ nos entrega el resto de dividir el número de años trabajados por 5, es decir, el número de años que han transcurrido desde el último quinquenio, y para calcular el número de años que faltan debemos restarle a 5 este número.

3) La cantidad acumulada a través de los años por concepto de quinquenios es

$$\text{CantAcum} = \text{CantInicial} / i * (\text{pow}(1+i, \text{NumQuin}) - 1);$$

es decir $\text{CantidadInicial} \cdot \frac{(1+i)^{\text{NumQuin}} - 1}{i}$, porque estamos frente a la suma de los términos de una progresión geométrica.

4) El mensaje correspondiente al número de años que faltan para completar un nuevo quinquenio, discrimina si falta 1 o más años.

15

Escriba un programa en C que solicite al usuario el ingreso de un número entero del 1 al 12 y muestre por pantalla el mes asociado a ese número y el número de días asociado al mes elegido. Por ejemplo, si el usuario ingresa 3 el programa debe mostrar el mensaje "MARZO tiene 31 días".

Si el usuario elige el número 2, entonces el programa debe mostrar el mensaje: "FEBRERO tiene 28 ó 29 días".

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int Mes;
    printf("\nIngrese un número entero del 1 al 12 : ",163);
    scanf("%i",&Mes);

    if(Mes==1) printf("\nENERO tiene 31 días\n",161);
    if(Mes==2) printf("\nFEBRERO tiene 28 ó 29 días\n",162,161);
    if(Mes==3) printf("\nMARZO tiene 31 días\n",161);
    if(Mes==4) printf("\nABRIL tiene 30 días\n",161);
    if(Mes==5) printf("\nMAYO tiene 31 días\n",161);
    if(Mes==6) printf("\nJUNIO tiene 30 días\n",161);
    if(Mes==7) printf("\nJULIO tiene 31 días\n",161);
    if(Mes==8) printf("\nAGOSTO tiene 31 días\n",161);
    if(Mes==9) printf("\nSEPTIEMBRE tiene 30 días\n",161);
    if(Mes==10) printf("\nOCTUBRE tiene 31 días\n",161);
    if(Mes==11) printf("\nNOVIEMBRE tiene 30 días\n",161);
    if(Mes==12) printf("\nDICIEMBRE tiene 31 días\n",161);

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida 1

```
Ingrese un número entero del 1 al 12 : 7
JULIO tiene 31 días
Presione una tecla para continuar . . .
```

Salida 2

```
Ingrese un número entero del 1 al 12 : 2
FEBRERO tiene 28 ó 29 días
Presione una tecla para continuar . . .
```

Salida 3

```
Ingrese un número entero del 1 al 12 : 5
MAYO tiene 31 días
Presione una tecla para continuar . . .
```

Salida 4

```
Ingrese un número entero del 1 al 12 : 9
SEPTIEMBRE tiene 30 días
Presione una tecla para continuar . . .
```


Salida 5

```
Ingrese un número entero del 1 al 12 : 15
Presione una tecla para continuar . . .
```

Observaciones:

1) En este ejercicio no se ocupa else porque existen muchas otras alternativas a las analizadas, es decir, el usuario podría ingresar equivocadamente 15 ó -3, y serían valores aceptables. Por ejemplo, lo contrario de `Mes==1` es `Mes` igual a cualquier valor entero distinto de 1.

2) Otra forma de resolver el mismo ejercicio es usando la estructura condicional SWITCH cuya sintaxis es:

```
switch ( expresion )
{
    declaraciones
    .
    .
    .
    case expresion_constante :
        sentencias que se ejecutan si expresion=expresion_constante
        .
        .
        .
    break;
    default :
```

sentencias que se ejecutan si expresion no es igual a expresion_constante

```
}
```

Usando la sentencia condicional anterior, se tiene que el programa podría ser:

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int Mes;
    printf("\nIngrese un número entero del 1 al 12 : ",163);
    scanf("%i",&Mes);

    switch(Mes)
    {
        case 1:
            printf("\nENERO tiene 31 días\n",161);
            break;
        case 2:
            printf("\nFEBRERO tiene 28 días\n",162,161);
            break;
        case 3:
            printf("\nMARZO tiene 31 días\n",161);
            break;
        case 4:
            printf("\nABRIL tiene 30 días\n",161);
            break;
        case 5:
            printf("\nMAYO tiene 31 días\n",161);
            break;
        case 6:
            printf("\nJUNIO tiene 30 días\n",161);
            break;
        case 7:
            printf("\nJULIO tiene 31 días\n",161);
            break;
        case 8:
            printf("\nAGOSTO tiene 31 días\n",161);
```

```

    break;
case 9:
    printf("\nSEPTIEMBRE tiene 30 d%cas\n",161);
    break;
case 10:
    printf("\nOCTUBRE tiene 31 d%cas\n",161);
    break;
case 11:
    printf("\nNOVIEMBRE tiene 30 d%cas\n",161);
    break;
case 12:
    printf("\nDICIEMBRE tiene 31 d%cas\n",161);
    break;
default:
    printf("\nHa ingresado un n%cmero de mes err%ceño\n",163,162);
    break;
}

printf("\n\n");
system("PAUSE");
return 0;
}


```

Las salidas de este programa son exactamente iguales a las del programa anterior, excepto en el caso de que el usuario ingrese un número de mes que no corresponda y en tal situación aparece un mensaje indicándoselo, a diferencia del programa anterior que nada indica.

```

Ingrese un número entero del 1 al 12 : 16
Ha ingresado un número de mes erróneo
Presione una tecla para continuar . . .

```



Escriba un programa que determine el menor de dos números enteros ingresados por el usuario.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{

    printf("\nIngrese los n%cmeros enteros a comparar\n",163);
    int Num1, Num2, Menor;

    printf("\nPrimer entero : ");
    scanf("%i",&Num1);

    printf("\nSegundo entero : ");
    scanf("%i",&Num2);

    if(Num1<Num2)
    {
        Menor=Num1;
        printf("\nEl menor entre %i y %i es %i\n",Num1,Num2,Menor);
    }
    else if(Num2<Num1)
    {
        Menor=Num2;
        printf("\nEl menor entre %i y %i es %i\n",Num1,Num2,Menor);
    }
    else
        printf("\nLos n%cmeros son iguales\n",163);

    printf("\n\n");
    system("PAUSE");
```

```
    return 0;  
}
```

Salida 1

```
Ingrese los números enteros a comparar  
Primer entero : 2  
Segundo entero : 3  
El menor entre 2 y 3 es 2  
Presione una tecla para continuar . . .
```

Salida 2

```
Ingrese los números enteros a comparar  
Primer entero : -7  
Segundo entero : -7  
Los números son iguales  
Presione una tecla para continuar . . .
```

Salida 3

```
Ingrese los números enteros a comparar  
Primer entero : 6  
Segundo entero : -2  
El menor entre 6 y -2 es -2  
Presione una tecla para continuar . . .
```

Observaciones:

1) No siempre la declaración de variables debe ir en las primeras líneas del programa principal; lo importante es siempre declarar algo antes de ocupar ese algo. En este ejercicio se declararon las variables Num1, Num2 y Menor después del mensaje por pantalla "Ingrese los números enteros a comparar".

```
printf("\nIngrese los n%cmeros enteros a comparar\n",163);  
int Num1, Num2, Menor;
```

2) Otra forma de resolver el mismo problema es usando el operador condicional ? cuya forma general es :

Condicion ? Expresion1 : Expresion2;

que es casi equivalente a

```
if(Condicion)  
    Expresion1;  
else  
    Expresion2;
```

Usando el operador anterior el programa podría ser:

```
#include <stdio.h>  
#include <stdlib.h>  
  
int main()  
{  
    printf("\nIngrese los n%cmeros enteros a comparar\n",163);  
    int Num1, Num2, Menor;  
  
    printf("\nPrimer entero : ");  
    scanf("%i",&Num1);  
  
    printf("\nSegundo entero : ");  
    scanf("%i",&Num2);  
  
    Menor= Num1<=Num2 ? Num1:Num2;  
  
    printf("\nEl menor entre %i y %i es %i\n",Num1,Num2,Menor);  
  
    printf("\n\n");  
    system("PAUSE");  
    return 0;  
}
```

cuya salida es:

```
Ingrese los números enteros a comparar
Primer entero : 3
Segundo entero : 2
El menor entre 3 y 2 es 2
Presione una tecla para continuar . . .
```



18

Escriba un programa en C que solicite al usuario el ingreso de tres números enteros y muestre por pantalla los mismos números, pero ordenados de menor a mayor.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int Num1, Num2, Num3;
    int Menor, Medio, Mayor;

    printf("\nIngrese tres números enteros\n",163);
    printf("\nPrimer número : ",163);
    scanf("%i",&Num1);
    printf("\nSegundo número : ",163);
    scanf("%i",&Num2);
    printf("\nTercer número : ",163);
    scanf("%i",&Num3);
```

```

if(Num1<=Num2 && Num2<=Num3)
{
    Menor=Num1; Medio=Num2; Mayor=Num3;
}

if(Num1<=Num3 && Num3<=Num2)
{
    Menor=Num1; Medio=Num3; Mayor=Num2;
}

if(Num2<=Num1 && Num1<=Num3)
{
    Menor=Num2; Medio=Num1; Mayor=Num3;
}

if(Num2<=Num3 && Num3<=Num1)
{
    Menor=Num2; Medio=Num3; Mayor=Num1;
}

if(Num3<=Num1 && Num1<=Num2)
{
    Menor=Num3; Medio=Num1; Mayor=Num2;
}

if(Num3<=Num2 && Num2<=Num1)
{
    Menor=Num3; Medio=Num2; Mayor=Num1;
}

printf("\nLos n%cmeros ordenados de menor a mayor son :",163);
printf("\n\n\t%i\t%i\t%i\n",Menor,Medio,Mayor);

printf("\n\n");
system("PAUSE");
return 0;
}

```


Salida 1

```
Ingrese tres números enteros
Primer número : -5
Segundo número : 3
Tercer número : -9
Los números ordenados de menor a mayor son :
    -9    -5    3
Presione una tecla para continuar . . .
```

Salida 2

```
Ingrese tres números enteros
Primer número : 2
Segundo número : 0
Tercer número : 0
Los números ordenados de menor a mayor son :
    0    0    2
Presione una tecla para continuar . . .
```

Salida 3

```
Ingrese tres números enteros
Primer número : 7
Segundo número : 3
Tercer número : -3
Los números ordenados de menor a mayor son :
    -3    3    7
Presione una tecla para continuar . . .
```

Observaciones:

1) Una forma de comparar tres números es considerar todas las posibles combinaciones de ordenación de los tres números; no es lo óptimo, pero resulta fácil de entender.

Si n_1 , n_2 y n_3 son los números, entonces las posibles combinaciones son, donde el número más a la izquierda es el menor y el que está más a la derecha el mayor:

1) n_1 n_2 n_3

2) n_1 n_3 n_2

3) n_2 n_1 n_3

4) n_2 n_3 n_1

5) n_3 n_1 n_2

6) n_3 n_2 n_1

2) Una versión ligeramente diferente del programa anterior es:

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int Num1, Num2, Num3;
    int Menor, Medio, Mayor;

    printf("\nIngrese tres n%cmeros enteros\n",163);
    printf("\nPrimer n%cmero : ",163);
    scanf("%i",&Num1);
    printf("\nSegundo n%cmero : ",163);
    scanf("%i",&Num2);
    printf("\nTercer n%cmero : ",163);
    scanf("%i",&Num3);

    if(Num1<=Num2 && Num2<=Num3)
    {
        Menor=Num1; Medio=Num2; Mayor=Num3;
```

```

}

else if(Num1<=Num3 && Num3<=Num2)
{
    Menor=Num1; Medio=Num3; Mayor=Num2;
}

else if(Num2<=Num1 && Num1<=Num3)
{
    Menor=Num2; Medio=Num1; Mayor=Num3;
}

else if(Num2<=Num3 && Num3<=Num1)
{
    Menor=Num2; Medio=Num3; Mayor=Num1;
}

else if(Num3<=Num1 && Num1<=Num2)
{
    Menor=Num3; Medio=Num1; Mayor=Num2;
}

else
{
    Menor=Num3; Medio=Num2; Mayor=Num1;
}

printf("\nLos n%cmeros ordenados de menor a mayor son :",163);
printf("\n\n\t%i\t%i\t%i\n",Menor,Medio,Mayor);

printf("\n\n");
system("PAUSE");
return 0;
} □

```

20

Escriba un programa en C que solicite al usuario el ingreso de las coordenadas de un punto del plano y luego muestre un mensaje por pantalla que indique el cuadrante al cual pertenece el punto o si tal punto está en alguno de los ejes cartesianos.

Además debe calcular y mostrar por pantalla la distancia del punto ingresado al origen, e indicar si tal distancia es menor que $\sqrt{3}$

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    float x, y, distancia;

    printf("\nIngrese las coordenadas de un punto del plano\n");
    printf("\nx = ");
    scanf("%f",&x);
    printf("\ny = ");
    scanf("%f",&y);

    if(x==0.0 || y==0.0)
        printf("\nEl punto est%c en uno de los ejes\n",160);
    if(x>0 && y>0)
        printf("\nEl punto est%c en el primer cuadrante\n",160);
    if(x<0 && y>0)
        printf("\nEl punto est%c en el segundo cuadrante\n",160);
    if(x<0 && y<0)
        printf("\nEl punto est%c en el tercer cuadrante\n",160);
    if(x>0 && y<0)
        printf("\nEl punto est%c en el cuarto cuadrante\n",160);
```

```

distancia=sqrt(x*x+y*y);

printf("\nLa distancia desde (%0.1f,%0.1f) al origen es %0.2f\n",x,y,distancia);

if(distancia<sqrt(3)) printf("\nLa distancia es menor que ra%cz de 3\n",161);

printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida 1

```

Ingrese las coordenadas de un punto del plano
x = -2
y = -6
El punto está en el tercer cuadrante
La distancia desde <-2.0,-6.0> al origen es 6.32
Presione una tecla para continuar . . .

```

Salida 2

```

Ingrese las coordenadas de un punto del plano
x = 1
y = -1
El punto está en el cuarto cuadrante
La distancia desde <1.0,-1.0> al origen es 1.41
La distancia es menor que raíz de 3
Presione una tecla para continuar . . .

```

Salida 3

```
Ingrese las coordenadas de un punto del plano
x = 0
y = -3
El punto está en uno de los ejes
La distancia desde <0.0,-3.0> al origen es 3.00
Presione una tecla para continuar . . .
```

Observaciones:

1) Un punto del plano está en uno de los ejes si alguna de sus componentes es cero.

Un punto está en el primer cuadrante si ambas coordenadas son positivas, está en el segundo cuadrante si la primera coordenada es negativa y la segunda positiva, está en el tercer cuadrante si ambas coordenadas son negativas, y está en el cuarto cuadrante cuando la primera coordenada es positiva y la segunda negativa.

2) Si alguna de las coordenadas del punto es cero, este programa se puede mejorar indicando en qué eje está; para ello recordemos que si la primera coordenada es cero, el punto está sobre el eje vertical, y si la segunda coordenada es cero está sobre el eje horizontal, y si ambas coordenadas son cero entonces el punto coincide con el origen. La codificación en este caso sería

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    float x, y, distancia;

    printf("\nIngrese las coordenadas de un punto del plano\n");
    printf("\nx = ");
    scanf("%f",&x);
    printf("\ny = ");
    scanf("%f",&y);
```

```

if(x==0.0)
    printf("\nEl punto est%c en el eje vertical\n",160);
if(y==0.0)
    printf("\nEl punto est%c en el eje horizontal\n",160);
if(x==0.0 && y==0.0)
    printf("\nEl punto es el origen\n");
if(x>0 && y>0)
    printf("\nEl punto est%c en el primer cuadrante\n",160);
if(x<0 && y>0)
    printf("\nEl punto est%c en el segundo cuadrante\n",160);
if(x<0 && y<0)
    printf("\nEl punto est%c en el tercer cuadrante\n",160);
if(x>0 && y<0)
    printf("\nEl punto est%c en el cuarto cuadrante\n",160);

distancia=sqrt(x*x+y*y);

printf("\nLa distancia desde (%0.1f,%0.1f) al origen es %0.2f\n",x,y,distancia);

if(distancia<sqrt(3)) printf("\nLa distancia es menor que ra%cz de 3\n",161);

printf("\n\n");
system("PAUSE");
return 0;
}

```

cuyas salidas son

Salida 1

```

Ingrese las coordenadas de un punto del plano
x = 0
y = 1
El punto está en el eje vertical
La distancia desde <0.0,1.0> al origen es 1.00
La distancia es menor que raíz de 3
Presione una tecla para continuar . . .

```

Salida 2

```
Ingrese las coordenadas de un punto del plano
x = 1
y = 0
El punto está en el eje horizontal
La distancia desde <1.0,0.0> al origen es 1.00
La distancia es menor que raíz de 3
Presione una tecla para continuar . . .
```

Salida 3

```
Ingrese las coordenadas de un punto del plano
x = 0
y = 0
El punto está en el eje vertical
El punto está en el eje horizontal
El punto es el origen
La distancia desde <0.0,0.0> al origen es 0.00
La distancia es menor que raíz de 3
Presione una tecla para continuar . . .
```



27

Escriba una aplicación en C que lea tres números enteros. Si el primero es negativo, entonces muestre por pantalla el producto de los tres números. En caso contrario, muestre la suma de los números.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int Num1, Num2, Num3;
    printf("\nIngrese primer número : ",163);
    scanf("%d",&Num1);
    printf("\nIngrese segundo número : ",163);
    scanf("%d",&Num2);
    printf("\nIngrese tercer número : ",163);
    scanf("%d",&Num3);

    if(Num1<0) printf("\nEl producto es %d\n",Num1*Num2*Num3);
    else printf("\nLa suma es %d\n",Num1+Num2+Num3);

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida 1

```
Ingrese primer número : 4
Ingrese segundo número : 3
Ingrese tercer número : 1
La suma es 8
Presione una tecla para continuar . . .
```

Salida 2

```
Ingrese primer número : -2
Ingrese segundo número : 3
Ingrese tercer número : 4
El producto es -24
Presione una tecla para continuar . . . 
```

28

Escriba un programa en C que solicite al usuario el ingreso de su edad y dependiendo de ésta muestra un mensaje apropiado en color, como se indica en la siguiente tabla:

Edad (años)	Mensaje	Color
0 a 1	Eres un bebé	Verde
2 a 15	Eres un niño	Azul
16 a 20	Eres un adolescente	Café
21 a 59	Eres un adulto	Rojo
60 o más	Eres de la tercera edad	Amarillo

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <conio2.h>

int main()
{
    int edad;

    printf("\nIngrese su edad : ");
    scanf("%i",&edad);
```

```

textbackground(WHITE);

if (edad>=0 && edad <=1)
{
    textcolor(GREEN);
    printf("\nEres un beb%c\n",130);
}
if (edad>=2 && edad <=15)
{
    textcolor(BLUE);
    printf("\nEres un ni%co\n",164);
}
if (edad>=16 && edad <=20)
{
    textcolor(BROWN);
    printf("\nEres un adolescente\n");
}
if (edad>=21 && edad <=59)
{
    textcolor(RED);
    printf("\nEres un adulto\n");
}
if (edad>=60)
{
    textcolor(YELLOW);
    printf("\nEres de la tercera edad\n");
}

printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida 1

```

Ingrese su edad : 4
Eres un niño
Presione una tecla para continuar . . .

```

Salida 2

```
Ingrese su edad : 54
Eres un adulto
Presione una tecla para continuar . . .
```

Salida 3

```
Ingrese su edad : 18
Eres un adolescente
Presione una tecla para continuar . . .
```

Observación:

Se asume que si el usuario tiene de 0 a 1 año, es un bebé muy aventajado, al igual que aquellos que tienen muy poca edad :)

30

Escriba un programa en C que determine si un año ingresado por el usuario es bisiesto o no.

(**Indicación:** Un año es bisiesto si es divisible por 4, pero no si es divisible por 100, a menos que también sea divisible por 400)

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main()
{
```

```

int agno;
printf("\nIngrese un año : ",164);
scanf("%i",&agno);

if((agno%4==0 && (agno%100!=0)) || agno%400==0)
    printf("\nEl año %i es bisiesto\n",164,agno);
else
    printf("\nEl año %i no es bisiesto\n",164,agno);

printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida 1

```

Ingrese un año : 2015
El año 2015 no es bisiesto
Presione una tecla para continuar . . .

```

Salida 2

```

Ingrese un año : 2016
El año 2016 es bisiesto
Presione una tecla para continuar . . .

```

Observaciones:

1) Un año es bisiesto si es divisible por 4, a menos que sea divisible por 100. Sin embargo, si un año es divisible por 100 y además es divisible por 400, entonces también será bisiesto.

2) Usando lógica proposicional, se tiene que si

$p =$ el año es divisible por 4

$q =$ el año es divisible por 100

r = el año es divisible por 400

entonces un año es bisiesto si la proposición $(p \wedge \neg q) \vee r$ es verdadera.

3) El operador relacional distinto (\neq) se escribe en C como $!=$, que se lee como "no igual". Recordemos que el símbolo $!$ se usa en C para negar algo.

4) Se usa el identificador `agno`, en desmedro de año, porque la letra ñ no es un caracter admisible para identificadores. \square

31

Escriba un programa que solicite al usuario el ingreso de la longitud de dos lados de un triángulo y del ángulo entre ellos expresado en grados, y calcule y muestre por pantalla la longitud del tercer lado y los otros dos ángulos, en grados, usando el teorema del coseno. El programa debe mostrar un mensaje en verde que indique si el triángulo es rectángulo isósceles. Note que C trabaja en radianes.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <conio2.h>
```

```
int main()
```

```
{
```

```
    float a, b, c, alfa, beta, gamma;
```

```
    printf("\nIngrese las longitudes de dos lados de un triángulo\n",160);
```

```
    printf("\nLongitud lado 1 : ");
```

```
    scanf("%f",&b);
```

```
    printf("\nLongitud lado 2 : ");
```

```
    scanf("%f",&c);
```

```
    printf("\nIngrese el ángulo, en grados, entre los lados anteriores : ",160);
```

```

scanf("%f",&alfa);

a=sqrt(b*b+c*c-2*b*c*cos(M_PI/180*alfa));

beta=acos((a*a+c*c-b*b)/(2*a*c));
gamma=acos((a*a+b*b-c*c)/(2*a*b));

printf("\n\tLado 3 : %0.2f\n",a);
printf("\n\tbeta : %0.2f%c\n",beta*180/M_PI,248);
printf("\n\tgamma : %0.2f%c\n\n",gamma*180/M_PI,248);

if((alfa==90.0 || beta==90.0 || gamma==90.0) && (a==b || a==c || b==c))
{
    textbackground(WHITE);
    textcolor(GREEN);
    printf("\nEl triángulo es rectángulo isósceles\n",160,160,162);
}

printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida 1

```

Ingrese las longitudes de dos lados de un triángulo
Longitud lado 1 : 4
Longitud lado 2 : 5
Ingrese el ángulo, en grados, entre los lados anteriores : 36.8698
    Lado 3 : 3.00
    beta : 53.13°
    gamma : 90.00°

Presione una tecla para continuar . . .

```

Salida 2

```
Ingrese las longitudes de dos lados de un triángulo
Longitud lado 1 : 2
Longitud lado 2 : 2
Ingrese el ángulo, en grados, entre los lados anteriores : 90
    Lado 3 : 2.83
    beta : 45.00°
    gamma : 45.00°

El triángulo es rectángulo isósceles

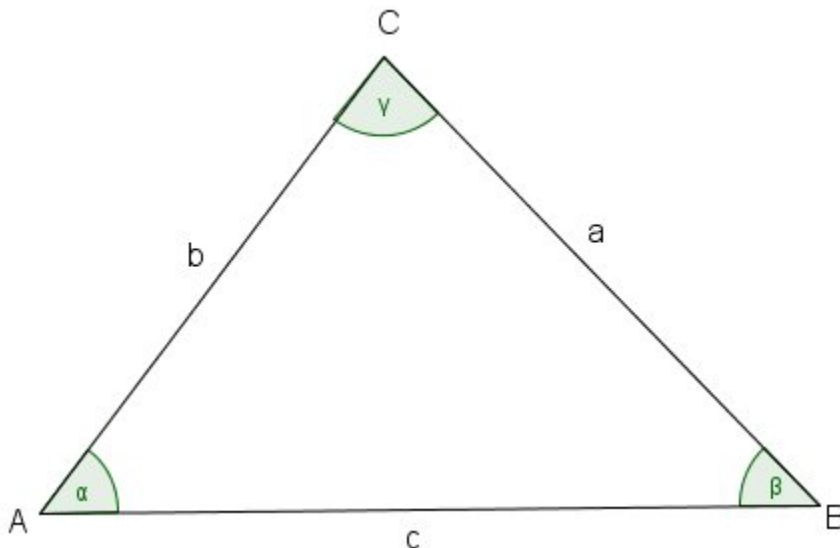
Presione una tecla para continuar . . .
```

Observaciones:

1) Un triángulo es rectángulo isósceles si alguno de sus ángulos interiores es igual a 90 grados y si además algún par de lados tienen igual longitud. Por lo anterior, la condición a considerar es:

$$(alfa==90.0 \ || \ beta==90.0 \ || \ gamma==90.0) \ \&\& \ (a==b \ || \ a==c \ || \ b==c)$$

2) TEOREMA DEL COSENO



En todo triángulo de ángulos α , β y γ , y lados opuestos respectivos a , b y c se cumple que:

$$(a) \ a^2 = b^2 + c^2 - 2bc \cos(\alpha)$$

$$(b) \ b^2 = a^2 + c^2 - 2ac \cos(\beta)$$

$$(c) \ c^2 = a^2 + b^2 - 2ab \cos(\gamma) \quad \square$$

32

Escriba un programa en C que pregunte el año actual y la edad de una persona y calcule la edad de esa persona en el año 2047.

Además se debe mostrar un mensaje por pantalla que indique si tal persona superará los 50 años en el 2047.

Asuma que la edad que se ingresa es la edad que se cumple en el año actual.

Solución:

Código Fuente

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
int main()
```

```
{
```

```
    int EdadActual, AgnoActual, EdadFutura;
```

```
    int AgnoFuturo=2047;
```

```
    printf("\nIngrese su edad en años : ",164);
```

```
    scanf("%i",&EdadActual);
```

```
    printf("\nIngrese el año actual : ",164);
```

```
    scanf("%i",&AgnoActual);
```

```
    EdadFutura=EdadActual+(AgnoFuturo-AgnoActual);
```

```
    printf("\nEn el año %i tendrá %i años en el %i",164,AgnoFuturo,160,EdadFutura,164);
```

```
if(EdadFutura>50)
    printf("\nUsted superar%c los 50 años el %i\n",160,164,AgnoFuturo);

printf("\n\n");
system("PAUSE");
return 0;
}
```

Salida 1

```
Ingrese su edad en años : 18
Ingrese el año actual : 2015
En el año 2047 tendrás 50 años
Presione una tecla para continuar . . .
```

Salida 2

```
Ingrese su edad en años : 12
Ingrese el año actual : 2018
En el año 2047 tendrás 41 años
Presione una tecla para continuar . . .
```



33

Escriba un programa en C que, haciendo uso de la orden SWITCH, muestre un menú que permita al usuario elegir entre las opciones siguientes: calcular el área y el perímetro de un círculo, rectángulo o triángulo. El usuario es quien ingresa el radio del círculo, el ancho y el alto del rectángulo, y los lados del triángulo. Muestre por pantalla los resultados.

Solución:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    int opcion;
    float radio, AreaCirc, PerimCirc;
    float ancho, largo, AreaRect, PerimRect;
    float lado1, lado2, lado3, AreaTriang, PerimTriang;
    float s, SubR;

    printf("\n\t1\t%crea y per%cmetro de un c%rculo\n",181,161,161);
    printf("\n\t2\t%crea y per%cmetro de un rect%ngulo\n",181,161,160);
    printf("\n\t3\t%crea y per%cmetro de un tri%ngulo\n",181,161,160);

    printf("\n\n\tIngrese una opci%cn (1, 2 %c 3) : ",162,162);
    scanf("%i",&opcion);

    switch(opcion)
    {
        case 1 :
            printf("\n\n");
            printf("\n\tIngrese el radio del c%rculo : ",161);
            scanf("%f",&radio);
            AreaCirc=M_PI*radio*radio;
            PerimCirc=2*M_PI*radio;
            printf("\n%crea del c%rculo : %0.2f\n",181, 161,AreaCirc);
            printf("\nPer%cmetro del c%rculo : %0.2f\n",161,161,PerimCirc);
            break;
        case 2 :
            printf("\n\n");
            printf("\n\tIngrese el ancho del rect%ngulo : ",160);
            scanf("%f",&ancho);
            printf("\n\tIngrese el alto del rect%ngulo : ",160);
            scanf("%f",&largo);
            AreaRect=ancho*largo;
```

```

PerimRect=2*(ancho+largo);
printf("\n%crea del rect%cngulo : %0.2f\n",181, 160,AreaRect);
printf("\nPer%cmetro del rect%cngulo : %0.2f\n",161,160,PerimRect);
break;
case 3 :
printf("\n\n");
printf("\nIngrese la longitud del primer lado del tri%cngulo : ",160);
scanf("%f",&lado1);
printf("\nIngrese la longitud del segundo lado del tri%cngulo : ",160);
scanf("%f",&lado2);
printf("\nIngrese la longitud del tercer lado del tri%cngulo : ",160);
scanf("%f",&lado3);
PerimTriang=lado1+lado2+lado3;
printf("\nPer%cmetro del tri%cngulo : %0.2f\n",161,160,PerimTriang);

s=PerimTriang/2;
SubR=s*(s-lado1)*(s-lado2)*(s-lado3);

if(SubR>=0)
{
AreaTriang=sqrt(SubR);
printf("\n%crea del tri%cngulo : %0.2f\n",181, 160,AreaTriang);
}
else
printf("\nNo se puede calcular el %crea\n",160);
break;
default:
printf("\nLa opci%n elegida puede ser 1, 2 %c 3",162,162);
break;
}

printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida 1

```
1      área y perímetro de un círculo
2      área y perímetro de un rectángulo
3      área y perímetro de un triángulo

Ingrese una opción <1, 2 ó 3> : 1

Ingrese el radio del círculo : 2
Área del círculo : 12.57
Perímetro del círculo : 12.57

Presione una tecla para continuar . . .
```

Salida 2

```
1      área y perímetro de un círculo
2      área y perímetro de un rectángulo
3      área y perímetro de un triángulo

Ingrese una opción <1, 2 ó 3> : 2

Ingrese el ancho del rectángulo : 3
Ingrese el alto del rectángulo : 2
Área del rectángulo : 6.00
Perímetro del rectángulo : 10.00

Presione una tecla para continuar . . .
```

Salida 3

```
1      área y perímetro de un círculo
2      área y perímetro de un rectángulo
3      área y perímetro de un triángulo

Ingrese una opción <1, 2 ó 3> : 3

Ingrese la longitud del primer lado del triángulo : 3
Ingrese la longitud del segundo lado del triángulo : 7
Ingrese la longitud del tercer lado del triángulo : 5
Perímetro del triángulo : 15.00
Área del triángulo : 6.50

Presione una tecla para continuar . . .
```

Salida 4

```
1      área y perímetro de un círculo
2      área y perímetro de un rectángulo
3      área y perímetro de un triángulo

Ingrese una opción <1, 2 ó 3> : 3

Ingrese la longitud del primer lado del triángulo : 2
Ingrese la longitud del segundo lado del triángulo : 5
Ingrese la longitud del tercer lado del triángulo : 9
Perímetro del triángulo : 16.00
No se puede calcular el área

Presione una tecla para continuar . . .
```

Observaciones:

1) Si r representa el radio de un círculo, entonces

$$Area = \pi r^2$$

$$Perim = 2\pi r$$

2) Si a y b representan las longitudes de los lados de un rectángulo, entonces

$$Area = ab$$

$$Perim = 2(a + b)$$

3) Si a , b y c representan las longitudes de los lados de un triángulo, entonces

$$Area = \sqrt{s(s-a)(s-b)(s-c)}$$

$$\text{con } s = \frac{a+b+c}{2}$$

$$Perim = a + b + c \quad \square$$

34

Prepara un programa en C que permita al usuario elegir una de las tres opciones siguientes, ingresando el usuario los datos que sean necesarios:

a) Dibujar un rectángulo de 3 caracteres de ancho por 10 de largo, usando asteriscos de color rojo.

b) Calcular y mostrar por pantalla $x \cdot y$, con x entero tal que $0 < x < 100$, y y real tal que $-10 < y < 5000$.

c) Calcular y mostrar por pantalla la suma de los primeros n términos de la progresión aritmética :

1, 4, 7, 10,

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <conio2.h>

int main()
{
    int opcion;
    int x; float y;
    int n, PrimerTerm, Diferencia;
    float suma;

    printf("\n\t1 Dibujar un rectángulo\n",160);
    printf("\n\t2 Calcular xy\n");
    printf("\n\t3 Suma primeros términos progresión aritmética\n",130,162,130);

    printf("\n\tIngrese opción (1, 2 o 3) : ",162,162);
    scanf("%i",&opcion);
    printf("\n\n");

    switch(opcion)
    {
        case 1 :
            textbackground(WHITE);
            textcolor(RED);
            printf("\n\t*****");
            printf("\n\t*      *");
            printf("\n\t*****\n\n");
            textcolor(LIGHTGRAY);
            break;
        case 2 :
            printf("\nIngrese x (1<x<100): ");
            scanf("%i",&x);
            if(x<=0 || x>=100)
            {
                printf("\nReingrese x (1<x<100) : ");
                scanf("%i",&x);
            }
            printf("\nIngrese y (-10<y<5000): ");
            scanf("%f",&y);
            if(y<=-10 || y>=5000)
            {
                printf("\nReingrese y (-10<y<5000) : ");
```



```

        scanf("%f",&y);
    }
    printf("\n%i * %0.2f = %0.2f\n",x,y,x*y);
    break;
case 3 :
    PrimerTerm=1; Diferencia=3;
    printf("\nIngrese n : ");
    scanf("%i",&n);
    suma=n*PrimerTerm+(n*(n-1))/2.0*Diferencia;
    printf("\nLa suma de los primeros %i t%crminos de la PA 1,4,7,10,... es %0.1f\n",n,130,suma);
    break;
default:
    printf("\nHa ingresado una opci%cn equivocada\n");
    break;
}

printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida 1

```

1 Dibujar un rectángulo
2 Calcular xy
3 Suma primeros términos progresión aritmética
Ingrese opción <1, 2 ó 3> : 1

*****
*           *
*****

Presione una tecla para continuar . . .

```

Salida 2

```
1 Dibujar un rectángulo
2 Calcular xy
3 Suma primeros términos progresión aritmética
Ingrese opción (1, 2 ó 3) : 2

Ingrese x (1<x<100): 4
Ingrese y (-10<y<5000): 7
4 * 7.00 = 28.00

Presione una tecla para continuar . . .
```

Salida 3

```
1 Dibujar un rectángulo
2 Calcular xy
3 Suma primeros términos progresión aritmética
Ingrese opción (1, 2 ó 3) : 3

Ingrese n : 4
La suma de los primeros 4 términos de la PA 1,4,7,10,... es 22.0

Presione una tecla para continuar . . .
```

Observaciones:

1) La sentencia `textcolor(LIGHTGRAY);` se usó para reestablecer el color por defecto.

2) Dada la progresión aritmética (PA) $a, a + d, a + 2d, \dots, a + (n - 1)d, \dots$ la suma de los primeros n términos es $S_n = na + \frac{n(n-1)}{2} d$

36

Escriba un programa en C que calcule y muestre por pantalla el promedio, la desviación estándar y el rango de 3 notas, del 1 al 7, ingresadas por el usuario.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    float n1, n2, n3, Prom, DesvEst;
    float menor, mayor, Rango;

    printf("\nIngrese las notas\n");
    printf("\nNota 1 : ");
    scanf("%f",&n1);
    printf("\nNota 2 : ");
    scanf("%f",&n2);
    printf("\nNota 3 : ");
    scanf("%f",&n3);

    Prom=(n1+n2+n3)/3;

    printf("\nEl promedio es %0.1f\n",Prom);

    DesvEst=sqrt(1/3.*(pow(n1-Prom,2)+pow(n2-Prom,2)+pow(n3-Prom,2)));

    if(n1>=n2)
        mayor=n1;
    else
        mayor=n2;
```

```

if(mayor>=n3)
    mayor=mayor;
else
    mayor=n3;

if(n1<=n2)
    menor=n1;
else
    menor=n2;

if(menor<=n3)
    menor=menor;
else
    menor=n3;

Rango=mayor-menor;

printf("\nLa desviación estándar es %0.2f\n",162,160,DesvEst);
printf("\nEl rango es %0.2f\n",Rango);

printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida 1

```

Ingrese las notas
Nota 1 : 3.2
Nota 2 : 5.8
Nota 3 : 4.3
El promedio es 4.4
La desviación estándar es 1.07
El rango es 2.60
Presione una tecla para continuar . . .

```

Salida 2

```
Ingrese las notas
Nota 1 : 5
Nota 2 : 5
Nota 3 : 5
El promedio es 5.0
La desviación estándar es 0.00
El rango es 0.00
Presione una tecla para continuar . . .
```

Observaciones:

1) Si $x_1, x_2, x_3, \dots, x_n$ representan n datos, entonces

$$\text{Promedio} = \frac{\sum_{i=1}^n x_i}{n} = \frac{x_1 + x_2 + x_3 + \dots + x_n}{n}$$

$$\text{DesvEst} = \frac{1}{n} \sum_{i=1}^n (x_i - \text{Promedio})^2$$

$$\text{Rango} = \text{maximo}\{x_1, x_2, x_3, \dots, x_n\} - \text{minimo}\{x_1, x_2, x_3, \dots, x_n\}$$

2) En el ejercicio se calcula la mayor nota, determinando primero la mayor entre n_1 y n_2 , y luego de obtenida la mayor entre las dos se compara con la tercera nota para obtener la mayor nota de todas. El procedimiento es análogo para obtener la menor nota de las tres.

39

Escriba un programa en C que muestre por pantalla las raíces de la ecuación $ax^2 + bx + c = 0$, donde a, b y c son números reales ingresados por el usuario. Considere todos los casos posibles.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    float a, b, c;
    float Discriminante;
    float x1, x2, x1Real, x1Complejo;

    printf("\nIngrese a : ");
    scanf("%f",&a);
    printf("\nIngrese b : ");
    scanf("%f",&b);
    printf("\nIngrese c : ");
    scanf("%f",&c);

    if(a==0.0 && b!=0.0)
        printf("\nExiste una soluci%cn que es %0.2f\n\n",162,-c/b);

    if(a==0.0 && b==0.0 && c!=0.0)
        printf("\nNo existe soluci%cn\n\n",162);

    if(a==0.0 && b==0.0 && c==0.0)
        printf("\nTodos los valores de x reales son soluci%cn\n\n",162);

    if(a!=0.0)
    {
        Discriminante=b*b-4*a*c;
        if(Discriminante>0.0)
        {
            x1=(-b+sqrt(Discriminante))/(2*a);
            x2=(-b-sqrt(Discriminante))/(2*a);
            printf("\nExisten dos soluciones reales\n\n");
            printf("x1 = %0.2f y x2 = %0.2f\n\n",x1,x2);
        }
        else if(Discriminante==0)
        {
            x1=-b/(2*a);
```

```

x2=x1;
printf("\nExiste una soluci%cn real repetida\n\n",162);
printf("x1 = %0.2f y x2 = %0.2f\n\n",x1,x2);
}
else
{
x1Real=-b/(2*a);
x1Complejo=sqrt(-Discriminante)/(2*a);
printf("\nExisten dos soluciones complejas conjugadas\n\n");
if(x1Complejo>0.0)
{
printf("x1 = %0.2f + %0.2f i",x1Real,x1Complejo);
printf(" y x2 = %0.2f - %0.2f i\n\n",x1Real,x1Complejo);
}
else
{
printf("x1 = %0.2f - %0.2f i",x1Real,-x1Complejo);
printf(" y x2 = %0.2f + %0.2f i\n\n",x1Real,-x1Complejo);
}
}
}
}

printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida 1

```

Ingrese a : 1
Ingrese b : 0
Ingrese c : -3
Existen dos soluciones reales
x1 = 1.73 y x2 = -1.73

Presione una tecla para continuar . . .

```

Salida 2

```
Ingrese a : 0
Ingrese b : 1
Ingrese c : 4
Existe una solución que es -4.00
Presione una tecla para continuar . . .
```

Salida 3

```
Ingrese a : 1
Ingrese b : 1
Ingrese c : 3
Existen dos soluciones complejas conjugadas
x1 = -0.50 + 1.66 i   y   x2 = -0.50 - 1.66 i
Presione una tecla para continuar . . .
```

Salida 4

```
Ingrese a : 0
Ingrese b : 0
Ingrese c : 7
No existe solución
Presione una tecla para continuar . . .
```

Observaciones:

1) Las raíces de la ecuación cuadrática $ax^2 + bx + c = 0$ son:

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

$$x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

2) El discriminante $b^2 - 4ac$ permite determinar si las raíces son reales e iguales, reales y distintas, o complejas conjugadas, de acuerdo a como se indica en la siguiente tabla:

Discriminante	Tipo de Raíces	Valores
Cero	Reales e iguales	$x_1 = x_2 = -\frac{b}{2a}$
Mayor que cero	Reales y distintas	$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$ $x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$
Menor que cero	Complejas conjugadas	$x_1 = \frac{-b}{2a} + \frac{\sqrt{-(b^2 - 4ac)}}{2a}i$ $x_2 = \bar{x}_1$



GUÍA DE PRÁCTICO N° 7

CONTADORES, ACUMULADORES DE SUMA Y DE PRODUCTO, Y ESTRUCTURAS ITERATIVAS

Teoría

Contadores, Acumuladores de Suma y de Producto, y Estructuras Iterativas

27 Contadores

Un contador es una variable cuyo valor aumenta o disminuye en una cierta cantidad constante, cada vez que se repite la sentencia donde aparece el contador.

Los contadores siempre deben ser inicializados, es decir, asignarles un valor inicial.

Sintaxis 1:

```
contador=valor_inicial;  
  
InicioCiclo  
contador=contador+ constante;  
FinCiclo
```

Sintaxis 2:

```
contador=valor_inicial;  
  
InicioCiclo  
contador=contador- constante;  
FinCiclo
```

T 28 Acumuladores de Suma y de Producto

Un acumulador es una variable que permite, como su nombre lo indica, acumular una cierta cantidad, a medida que se repite la sentencia en donde aparece el acumulador. Si lo que se acumula es mediante sumas, entonces el acumulador es de suma, y si lo que se acumula es mediante productos, entonces el acumulador es de producto.

Los acumuladores deben ser inicializados. Es común inicializar el acumulador de suma en cero, y el acumulador de producto en uno.

Sintaxis Acumulador de Suma:

```
acumulador_de_suma=valor_inicial;  
  
InicioCiclo  
acumulador_de_suma=acumulador_de_suma+ x;  
FinCiclo
```

Sintaxis Acumulador de Producto:

```
acumulador_de_producto=valor_inicial;  
  
InicioCiclo  
acumulador_de_producto=acumulador_de_producto*x;  
FinCiclo
```

T 29 Estructuras Iterativas o de Repetición o Bucle

A veces es necesario ejecutar una instrucción o un bloque de instrucciones más de una vez, y para ello se utiliza una estructura iterativa, que permite repetir un proceso un número de veces determinado por alguna condición.

Las estructuras iterativas disponibles en C son while, do ...while y for

T 30 Estructura Iterativa While (Mientras)

La estructura iterativa while permite repetir una serie de acciones, mientras se cumpla una condición que se evalúa al inicio del bucle.

Sintaxis 1:

```
while(condicion) sentencia;
```

En este caso, mientras condicion sea verdadera se ejecuta sentencia.

Sintaxis 2:

```
while(condicion) { bloque de sentencias }
```

En este caso, mientras condicion sea verdadera se ejecuta el bloque de sentencias.

T 31 Estructura Iterativa Do ...While (Hacer ... Mientras)

La estructura iterativa do ... while permite repetir una serie de acciones, mientras se cumpla una condición que se evalúa al final del bucle.

Sintaxis 1:

```
do sentencia while(condicion);
```

En este caso, se ejecuta sentencia mientras condicion sea verdadera.

Sintaxis 2:

```
do { bloque de sentencias } while(condicion);
```

En este caso, se ejecuta el bloque de sentencias mientras condicion sea verdadera.

T 32 Estructura Iterativa For (Para)

La estructura iterativa for permite repetir una serie de acciones, mediante el uso de un contador, indicando el número de repeticiones a través de la utilización de una condición de inicio (expresion1), otra condición de término (expresion2) y la acción después de cada ciclo (expresion3).

Es común usar el bucle for cuando es conocido el número de repeticiones.

Sintaxis 1:

```
for(expresion1;expresion2;expresion3)  
sentencia;
```

En este caso, se ejecuta sentencia desde lo que indica expresion1 hasta lo que indica expresion2, incrementando o decrementando el contador según lo indica la expresion3

Sintaxis 2:

```
for(expresion1;expresion2;expresion3)  
{bloque de sentencias}
```

En este caso, se ejecuta bloque de sentencias desde lo que indica expresion1 hasta lo que indica expresion2, incrementando o decrementando el contador según lo indica la expresion3

Ejercicios

Contadores, Acumuladores de Suma y de Producto, y Estructuras Iterativas

[1] Escriba un programa en C que muestre por pantalla 100 veces el mensaje "Es mejor callar y parecer ignorante, que hablar y confirmarlo"

[2] Escriba un programa que calcule el cociente de dos números enteros ingresados por el usuario. El programa debe obligar al usuario a ingresar un denominador distinto de cero para calcular el cociente.

[3] Escriba un programa en C que muestre por pantalla todos los números múltiplos de 6 desde el 6 al 1800.

[4] Escriba un programa en C que muestre por pantalla los números del 1 al 100 en filas de 10 números. Cada fila debe aparecer con un color diferente.

[5] Escriba un programa en C que solicite al usuario el ingreso de las estaturas de 5 alumnos, cuyas alturas no deben superar los 2 metros, para luego calcular y mostrar por pantalla la estatura promedio.

[6] Use C para solicitar al usuario el ingreso de un número cardinal y luego mostrar por pantalla su tabla de multiplicar. Se debe repetir el proceso hasta que el usuario ingrese -1 .

[7] Escriba un programa en C que muestre por pantalla 100 veces el mensaje: "Debo estudiar todos los días"
Use color.

[8] Prepare un programa en C que permita calcular y mostrar por pantalla el cociente $(\frac{x}{y})$ de dos números x y y ingresados por el usuario, tales que $|x| < 1$ y $|y| > 1$

[9] Escriba un programa en C que dibuje n líneas horizontales de longitud L , donde se le exige al usuario el ingreso de un n tal que $2 \leq n \leq 23$ y de un entero L tal que $1 < L < n$.

[10] Escriba un programa en C que genere y muestre por pantalla 20 puntos del plano pertenecientes a la curva de ecuación $x^2 - y = 1$

[11] Escriba un programa en C que calcule y muestre por pantalla la suma de los dígitos de un número entero de 4 dígitos ingresado por el usuario. El programa debe obligar al usuario a ingresar un número de 4 dígitos

[12] Escriba un programa en C que calcule y muestre por pantalla, con 1 decimal, el promedio ponderado de 5 notas de un alumno. El usuario es quien ingresa las notas y las ponderaciones correspondientes como porcentaje (por ejemplo, 20 ó 25).

El programa además debe obligar al usuario a ingresar notas entre 1 y 7, inclusive.

El programa debe mostrar un mensaje, en color verde brillante, indicando si el alumno aprobó (promedio ponderado ≥ 4.0) o si reprobó (promedio ponderado < 4.0)

[13] Escriba un programa en C que calcule y muestre por pantalla el área $A = \sqrt{s(s-a)(s-b)(s-c)}$ de un triángulo, donde la longitud de los lados a , b y c es ingresada por el usuario, con $s = \frac{a+b+c}{2}$. El programa debe obligar al usuario a ingresar valores de a , b y c tales que la cantidad subradical $s(s-a)(s-b)(s-c)$ sea no negativa.

[14] Escriba un programa en C que calcule y muestre por pantalla el cociente de dos números enteros mayores o iguales que -3 y menores o iguales que 15 , ingresados por el usuario.

El programa debe mostrar por pantalla un mensaje indicando si se está dividiendo por cero.

[15] Prepare una aplicación en C, para $\alpha = 0, \frac{\pi}{16}, \frac{\pi}{8}, \dots, 2\pi$, que muestre por pantalla una tabla para cada una de las siguientes cónicas

a) La circunferencia

$$\begin{aligned}x &= r \cos(\alpha) \\y &= r \sin(\alpha)\end{aligned}$$

donde $r > 0$ es ingresado por el usuario

b) La elipse

$$\begin{aligned}x &= a \cos(\alpha) \\y &= b \sin(\alpha)\end{aligned}$$

donde $a > b$ y ambas son positivas e ingresadas por el usuario.

[16] Escriba un programa en C que calcule la suma de los pares que van desde 0 hasta N , donde N es un número natural par, ingresado por el usuario. Incluya en la suma el 0 y el N .

[17] Escriba un programa en C que solicite al usuario el ingreso de dos números enteros distintos, y exhiba en la consola el número total de pares entre los números ingresados incluyéndolos; además el programa debe mostrar por pantalla la suma de los números múltiplos de 3 que están entre los números ingresados sin considerarlos.

[18] Escriba un programa en C que calcule y muestre por pantalla, con 3 decimales, el resultado de

a) $\sum_{i=6}^{60} (-1)^i \cdot i$

b) $\prod_{n=1}^{50} \frac{n}{n+1}$

[19] Escriba un programa en C que solicite al usuario el ingreso de dos números enteros distintos, y exhiba en la consola el número total de impares entre los números ingresados incluyéndolos; además el programa debe mostrar por pantalla la suma de los números pares múltiplos de 7 que están entre los números ingresados sin considerarlos.

[20] Escriba un programa en C que calcule y muestre por pantalla, con 3 decimales, la suma $\sum_{i=1}^n \frac{1}{i(i+1)}$ donde n es un número natural ingresado por el usuario.

[21] Escriba un programa en C que muestre por pantalla la raíz de la ecuación $ax + b = 0$, donde a y b son números reales no nulos ingresados por el usuario.

[22] Use C para mostrar por pantalla la suma de los números pares del 2 al 200.

[23] Use C para calcular y mostrar por pantalla el resultado de:

a) $3 + \sum_{i=3}^{100} \frac{1}{i}$

b) $\prod_{k=2}^7 \frac{k+1}{k}$

[24] Use C para mostrar por pantalla los números pares del 2 al 200 en líneas de 10 números. Después de la última línea debe aparecer el mensaje "FIN DEL PROGRAMA" en color magenta.

[25] Escriba un programa en C que calcule y muestre por pantalla la distancia desde un punto P a una recta en el espacio.

El programa debe obligar al usuario a ingresar las coordenadas del punto P en el primer octante.

El usuario además ingresa las coordenadas del punto de la recta y del vector director.

[26] Escriba un programa en C que calcule y muestre por pantalla la suma de los primeros n números pares, donde n es un número natural ingresado por el usuario.

[27] Escriba un programa en C que solicite al usuario el ingreso de números enteros positivos y muestre por pantalla aquellos que sean pares y la suma de los impares.

Para indicar el fin del proceso el usuario debe ingresar un cero.

[28] Escriba un programa que calcule la suma de los primeros n números impares positivos, donde n es un número natural ingresado por el usuario.

[29] Escriba una aplicación de consola en C que calcule y muestre por pantalla en color rojo brillante, la suma de los primeros n múltiplos de 7, donde n es un número natural menor o igual a 15 ingresado por el usuario.

[30] Use C para calcular y mostrar por pantalla $\sum_{i=1}^{25} \frac{(-2)^i}{i}$ y $15 - \prod_{k=0}^7 \frac{1}{(k+2)}$

[31] Escriba un programa en C que muestre por pantalla con 4 decimales :

a) $50 + \sum_{k=5}^{100} \frac{k+1}{k}$

b) $\prod_{k=10}^{50} (k - 5)$

[32] Escriba un programa en C que muestre por pantalla las sumas parciales $S_n = \sum_{i=1}^n \frac{1}{i}$ donde n es un número entero positivo ingresado por el usuario.

[33] Escriba un programa en C que muestre por pantalla las sumas parciales $S_n = \sum_{i=1}^n \frac{1}{i}$ y los productos parciales $P_n = \prod_{i=1}^n \frac{1}{i}$, donde n es un número entero ingresado por el usuario.

[34] Escriba un programa en C que calcule la suma $\sum_{i=m}^n \frac{n-m+i}{i}$ donde m y n son ingresados por el usuario, con $m \leq n$.

[35] Escriba un programa en C que calcule y muestre por pantalla, con 9 decimales, las siguientes sumas:

a) $\sum_{n=-1}^{500} \frac{1}{(n^2+2)}$ (Rpta. : 1.692363438)

b) $4 - \sum_{k=-3}^{10} \frac{1}{2^k}$ (Rpta. : -11.99902344)

[36] Use C para solicitar al usuario el ingreso de 21 números enteros positivos y luego mostrar por pantalla la suma de los impares múltiplos de 5. El programa debe obligar al usuario a ingresar números enteros positivos.

[37] Escriba un programa en C que solicite al usuario el ingreso de un número real y luego evalúe en tal número la función

$$f(x) = \begin{cases} \sum_{i=1}^{50} \frac{1}{x^i} & , \text{ si } x > \sqrt{3} \\ \prod_{k=2}^6 (x - k) & , \text{ si } x \leq \sqrt{3} \end{cases}$$

mostrando por pantalla, con 2 decimales, la función evaluada en el punto ingresado por el usuario.

[38] Use C para determinar para qué valor de n la productoria $\prod_{z=1}^n \frac{z+5}{z+1}$ es mayor o igual a 2013.

PROBLEMAS RESUELTOS CONTADORES, ACUMULADORES DE SUMA Y DE PRODUCTO, Y ESTRUCTURAS ITERATIVAS

1

Escriba un programa en C que muestre por pantalla 100 veces el mensaje "Es mejor callar y parecer ignorante, que hablar y confirmarlo"

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int i;

    for(i=1;i<=100;i++)
        printf("\nEs mejor callar y parecer ignorante, que hablar y confirmarlo");

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```


2

Escriba un programa que calcule el cociente de dos números enteros ingresados por el usuario. El programa debe obligar al usuario a ingresar un denominador distinto de cero para calcular el cociente.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int Num, Den;
    printf("\nIngrese numerador : ");
    scanf("%i",&Num);
    do
    {
        printf("\nIngrese denominador, que debe ser distinto de cero : ");
        scanf("%i",&Den);
    }
    while(Den==0);

    printf("\n%i / %i = %f\n",Num, Den,(float) Num/Den);

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida 1

```
Ingrese numerador : 5
Ingrese denominador, que debe ser distinto de cero : 7
5 / 7 = 0.714286
Presione una tecla para continuar . . .
```

Salida 2

```
Ingrese numerador : 5
Ingrese denominador, que debe ser distinto de cero : 0
Ingrese denominador, que debe ser distinto de cero : 3
5 / 3 = 1.666667
Presione una tecla para continuar . . .
```

Observaciones:

1) La estructura iterativa `do { proceso } while(condicion);` permite repetir un proceso mientras condicion sea verdadera. En este caso proceso se repetirá al menos una vez.

Otra estructura iterativa muy similar es: `while(condicion) { proceso }` que permite repetir proceso mientras condicion sea verdadera. La diferencia con la estructura anterior es que proceso podría no ejecutarse si condicion es falsa de entrada.

2) Es común utilizar la estructura `do ... while` para solicitar el reingreso de información porque no satisface cierta condicion, como en el ejercicio en donde se le solicita al usuario reingresar el valor del denominador si éste es cero.

3)
$$Cuociente = \frac{Numerador}{Denominador}$$

4) En la sentencia `printf("\n%i / %i = %f\n", Num, Den, (float) Num/Den);` es necesario el uso de la conversión de tipo (casting) para indicarle al compilador que el cuociente de los números enteros Num y Den entrega como

resultado un número real (float); si no se incluye la conversión de tipo, entonces el resultado del cociente sería sólo la parte entera.

A continuación aparece el programa sin la conversión de tipo:

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int Num, Den;
    printf("\nIngrese numerador : ");
    scanf("%i",&Num);
    do
    {
        printf("\nIngrese denominador, que debe ser distinto de cero : ");
        scanf("%i",&Den);
    }
    while(Den==0);

    printf("\n%i / %i = %f\n",Num, Den, Num/Den);

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

cuya salida es:

```
Ingrese numerador : 5
Ingrese denominador, que debe ser distinto de cero : 3
5 / 3 = 0.000000
Presione una tecla para continuar . . .
```

Como se observa, el resultado es incorrecto porque se produce una incompatibilidad de tipos, pues el cociente de dos números enteros es un entero, pero se usa un especificador de formato %f que no corresponde. A continuación se muestra el mismo programa, pero cambiando el especificador de formato a %i

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int Num, Den;
    printf("\nIngrese numerador : ");
    scanf("%i",&Num);
    do
    {
        printf("\nIngrese denominador, que debe ser distinto de cero : ");
        scanf("%i",&Den);
    }
    while(Den==0);

    printf("\n%i / %i = %i\n",Num, Den, Num/Den);

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

cuya salida es:

```
Ingrese numerador : 5
Ingrese denominador, que debe ser distinto de cero : 3
5 / 3 = 1
Presione una tecla para continuar . . .
```



3

Escriba un programa en C que muestre por pantalla todos los números múltiplos de 6 desde el 6 al 1800.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int k;

    for(k=6;k<=1800;k=k+6)
        printf("\t%i",k);

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida

```
60      6      12      18      24      30      36      42      48      54
120     66     72     78     84     90     96     102    108    114
180     126    132    138    144    150    156    162    168    174
240     186    192    198    204    210    216    222    228    234
300     246    252    258    264    270    276    282    288    294
360     306    312    318    324    330    336    342    348    354
360     366    372    378    384    390    396    402    408    414
420     426    432    438    444    450    456    462    468    474
480     486    492    498    504    510    516    522    528    534
540     546    552    558    564    570    576    582    588    594
600     606    612    618    624    630    636    642    648    654
660     666    672    678    684    690    696    702    708    714
720     726    732    738    744    750    756    762    768    774
780     786    792    798    804    810    816    822    828    834
840     846    852    858    864    870    876    882    888    894
900     906    912    918    924    930    936    942    948    954
960     966    972    978    984    990    996    1002   1008   1014
1020    1026   1032   1038   1044   1050   1056   1062   1068   1074
1080    1086   1092   1098   1104   1110   1116   1122   1128   1134
1140    1146   1152   1158   1164   1170   1176   1182   1188   1194
1200    1206   1212   1218   1224   1230   1236   1242   1248   1254
1260    1266   1272   1278   1284   1290   1296   1302   1308   1314
1320    1326   1332   1338   1344   1350   1356   1362   1368   1374
1380    1386   1392   1398   1404   1410   1416   1422   1428   1434
1440    1446   1452   1458   1464   1470   1476   1482   1488   1494
1500    1506   1512   1518   1524   1530   1536   1542   1548   1554
1560    1566   1572   1578   1584   1590   1596   1602   1608   1614
1620    1626   1632   1638   1644   1650   1656   1662   1668   1674
1680    1686   1692   1698   1704   1710   1716   1722   1728   1734
1740    1746   1752   1758   1764   1770   1776   1782   1788   1794
1800

Presione una tecla para continuar . . .
```

Observaciones:

1) El programa anterior se basa en el hecho que si partimos con 6 y cada vez le sumamos 6 más, hasta llegar a 1800, entonces obtenemos todos los múltiplos de 6 solicitados.

Notemos que la expresión $k=k+6$ se lee "el valor actual de k es igual al valor anterior de k más 6", con esto logramos que k aumente en 6 unidades en cada iteración.

2) Otra forma de resolver el mismo problema es usando el hecho que $6k$, $k = 1, 2, \dots, 300$ me entrega los mismos múltiplos de 6 solicitados. El programa en este caso es:

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main()
{
```

```

int k;

for(k=1;k<=300;k++)
    printf("\t%i",6*k);

printf("\n\n");
system("PAUSE");
return 0;
}

```

cuya salida es:

60	66	72	78	84	90	96	102	108	114
120	126	132	138	144	150	156	162	168	174
180	186	192	198	204	210	216	222	228	234
240	246	252	258	264	270	276	282	288	294
300	306	312	318	324	330	336	342	348	354
360	366	372	378	384	390	396	402	408	414
420	426	432	438	444	450	456	462	468	474
480	486	492	498	504	510	516	522	528	534
540	546	552	558	564	570	576	582	588	594
600	606	612	618	624	630	636	642	648	654
660	666	672	678	684	690	696	702	708	714
720	726	732	738	744	750	756	762	768	774
780	786	792	798	804	810	816	822	828	834
840	846	852	858	864	870	876	882	888	894
900	906	912	918	924	930	936	942	948	954
960	966	972	978	984	990	996	1002	1008	1014
1020	1026	1032	1038	1044	1050	1056	1062	1068	1074
1080	1086	1092	1098	1104	1110	1116	1122	1128	1134
1140	1146	1152	1158	1164	1170	1176	1182	1188	1194
1200	1206	1212	1218	1224	1230	1236	1242	1248	1254
1260	1266	1272	1278	1284	1290	1296	1302	1308	1314
1320	1326	1332	1338	1344	1350	1356	1362	1368	1374
1380	1386	1392	1398	1404	1410	1416	1422	1428	1434
1440	1446	1452	1458	1464	1470	1476	1482	1488	1494
1500	1506	1512	1518	1524	1530	1536	1542	1548	1554
1560	1566	1572	1578	1584	1590	1596	1602	1608	1614
1620	1626	1632	1638	1644	1650	1656	1662	1668	1674
1680	1686	1692	1698	1704	1710	1716	1722	1728	1734
1740	1746	1752	1758	1764	1770	1776	1782	1788	1794
1800									

Presione una tecla para continuar . . .

que, como se observa, es exactamente igual a la salida del programa anterior.

4

Escriba un programa en C que muestre por pantalla los números del 1 al 100 en filas de 10 números. Cada fila debe aparecer con un color diferente.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <conio2.h>

int main()
{
    int i;
    int k=2;

    for(i=1;i<=100;i=i+1)
    {
        if(i==1) textcolor(2);
        if(i%10==0)
        {
            k++;
            printf(" %i\n",i);
        }
        else
            printf(" %i",i);

        textcolor(k);
    }
    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida

1	2	3	4	5	6	7	8	9	10		
11	12	13	14	15	16	17	18	19	20		
21	22	23	24	25	26	27	28	29	30		
31	32	33	34	35	36	37	38	39	40		
41	42	43	44	45	46	47	48	49	50		
51	52	53	54	55	56	57	58	59	60		
61	62	63	64	65	66	67	68	69	70		
71	72	73	74	75	76	77	78	79	80		
81	82	83	84	85	86	87	88	89	90		
91	92	93	94	95	96	97	98	99	100		

Presione una tecla para continuar . . .

Observaciones:

1) Para determinar si se ha completado la fila de los 10 números se analiza qué pasa con el valor de i actual, para ello se determina si el valor de i es múltiplo de 10, y en tal situación se salta a la siguiente línea.

2) Los colores, como se mencionó en un capítulo anterior, están asociados a valores numéricos, y podemos usar tal hecho para cambiar los colores gracias a la utilización de un contador k que se inicializa en 2 (color verde) y se aumenta de 1 en 1 a medida que se completa una línea de 10 números.

6

Use C para solicitar al usuario el ingreso de un número cardinal y luego mostrar por pantalla su tabla de multiplicar. Se debe repetir el proceso hasta que el usuario ingrese -1 .

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
```

```

int main()
{
    int num=0, i;

    while(num!=-1)
    {
        printf("\n");
        printf("\nIngrese un número cardinal (0,1,2,...) : ",163);
        scanf("%i",&num);

        if(num== -1) break;

        for(i=1;i<=12;i++)
            printf("\n%i * %i = %i",num,i,num*i);
    }

    printf("\n\n");
    system("PAUSE");
    return 0;
}

```

Salida 1

```

Ingrese un número cardinal <0,1,2,...> : 4
4 * 1 = 4
4 * 2 = 8
4 * 3 = 12
4 * 4 = 16
4 * 5 = 20
4 * 6 = 24
4 * 7 = 28
4 * 8 = 32
4 * 9 = 36
4 * 10 = 40
4 * 11 = 44
4 * 12 = 48

Ingrese un número cardinal <0,1,2,...> : -1

Presione una tecla para continuar . . .

```


Salida 2

```
Ingrese un número cardinal <0,1,2,....> : 9
9 * 1 = 9
9 * 2 = 18
9 * 3 = 27
9 * 4 = 36
9 * 5 = 45
9 * 6 = 54
9 * 7 = 63
9 * 8 = 72
9 * 9 = 81
9 * 10 = 90
9 * 11 = 99
9 * 12 = 108

Ingrese un número cardinal <0,1,2,....> : 5
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50
5 * 11 = 55
5 * 12 = 60

Ingrese un número cardinal <0,1,2,....> : -1

Presione una tecla para continuar . . .
```

Observaciones:

- 1) En el código fuente se ha inicializado la variable num en cero, porque hemos usado la estructura iterativa while(condicion) ... que primero revisa si condicion es verdadera y si eso no pasa, entonces el proceso no se ejecuta ni siquiera una vez.
- 2) Se ha incluido la línea de código `if(num==-1) break;` para salir del while y de ese modo terminar la ejecución del programa en caso que num sea igual a `-1`; si no se incluye tal línea aparecerá la tabla del `-1` y luego terminará el programa.
- 3) Se han considerado los productos hasta el 12 aunque el enunciado no lo indica.

8

Prepare un programa en C que permita calcular y mostrar por pantalla el cociente ($\frac{x}{y}$) de dos números x y y ingresados por el usuario, tales que $|x| < 1$ y $|y| > 1$

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    float x,y;

    do
    {
        printf("\nIngrese x : ");
        scanf("%f",&x);
    }
    while(x<=-1 || x>=1);

    do
    {
        printf("\nIngrese y : ");
        scanf("%f",&y);
    }
    while(y<=1 && y>=-1);

    printf("\n%0.2f / %0.2f = %0.2f\n",x,y,x/y);

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida

```
Ingrese x : 2
Ingrese x : 0.8
Ingrese y : 0
Ingrese y : -2
0.80 / -2.00 = -0.40

Presione una tecla para continuar . . .
```

Observaciones:

1) Tenemos que:

$$|x| < 1 \Leftrightarrow -1 < x < 1 \Leftrightarrow x > -1 \wedge x < 1$$

$$|y| > 1 \Leftrightarrow y > 1 \vee y < -1$$

Si negamos las expresiones anteriores obtenemos

$$(x > -1 \wedge x < 1)' = x \leq -1 \vee x \geq 1$$

$$(y > 1 \vee y < -1)' = y \leq 1 \wedge y \geq -1$$

que son justamente las condiciones que aparecen en las sentencias `do ... while`

2) La expresión `while(y<=1 && y>=-1)` es equivalente a `while(-1<=y && y<=1)`, pero no iguales a la expresión `while(-1<=y<=1)`

9

Escriba un programa en C que dibuje n líneas horizontales de longitud L , donde se le exige al usuario el ingreso de un n tal que $2 \leq n \leq 23$ y de un entero L tal que $1 < L < n$.

Solución:

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int n, L, i, k;

    do
    {
        printf("\nIngrese n (2<=n<=23) : ");
        scanf("%i",&n);
    }
    while(n<2 || n>23);

    do
    {
        printf("\nIngrese L (1<L<=n) : ",n);
        scanf("%i",&L);
    }
    while(L<=1 || L>=n);

    for(i=1;i<=n;i++)
    {
        printf("\n");
        for(k=1;k<=L;k++)
            printf("%c",196);
    }

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```



```

int main()
{
    int i;

    for(i=1;i<=20;i++)
        printf("\n(%i,%i)",i,i*i-1);

    printf("\n\n");
    system("PAUSE");
    return 0;
}

```

Salida

```

<1,0>
<2,3>
<3,8>
<4,15>
<5,24>
<6,35>
<7,48>
<8,63>
<9,80>
<10,99>
<11,120>
<12,143>
<13,168>
<14,195>
<15,224>
<16,255>
<17,288>
<18,323>
<19,360>
<20,399>

Presione una tecla para continuar . . .

```

Observaciones:

- 1) Un punto del plano (x, y) pertenece a la curva de ecuación $x^2 - y = 1$ si tiene la forma $(x, x^2 - 1)$
- 2) En el ejercicio hemos considerado los primeros 20 números naturales para x , y en esos valores hemos evaluado la función $y = x^2 - 1$
- 3) Este ejercicio podría haberse resuelto generando cualquier conjunto de valores, por ejemplo, valores reales que comienzan en 4 y van aumentando de 0.5 en 0.5 hasta completar los 20 pares ordenados, cuya codificación es:

```

#include <stdio.h>
#include <stdlib.h>

int main()
{
    float i;

    for(i=4;i<20*0.5+4;i=i+0.5)
        printf("\n(%0.1f , %0.2f)",i,i*i-1);

    printf("\n\n");
    system("PAUSE");
    return 0;
}

```

y cuya salida es:

```

<4.0 , 15.00>
<4.5 , 19.25>
<5.0 , 24.00>
<5.5 , 29.25>
<6.0 , 35.00>
<6.5 , 41.25>
<7.0 , 48.00>
<7.5 , 55.25>
<8.0 , 63.00>
<8.5 , 71.25>
<9.0 , 80.00>
<9.5 , 89.25>
<10.0 , 99.00>
<10.5 , 109.25>
<11.0 , 120.00>
<11.5 , 131.25>
<12.0 , 143.00>
<12.5 , 155.25>
<13.0 , 168.00>
<13.5 , 181.25>

Presione una tecla para continuar . . .

```

4) El índice que se usa en un ciclo for puede ser también una variable de tipo real.

11

Escriba un programa en C que calcule y muestre por pantalla la suma de los dígitos de un número entero de 4 dígitos ingresado por el usuario. El programa debe obligar al usuario a ingresar un número de 4 dígitos

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    int Num, dig1, dig2, dig3, dig4, suma;
    do
    {
        printf("\nIngrese un número entero de 4 dígitos : ",163,161);
        scanf("%i",&Num);
    }
    while(floor(1+log10(abs(Num)))!=4);

    dig1=Num%10;
    dig2=(Num/10)%10;
    dig3=(Num/100)%10;
    dig4=(Num/1000)%10;

    suma=dig1+dig2+dig3+dig4;

    printf("\nLa suma de los dígitos de %i es %i\n",161,Num,suma);

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```


Salida

```
Ingrese un número entero de 4 dígitos : 34
Ingrese un número entero de 4 dígitos : 2
Ingrese un número entero de 4 dígitos : 137
Ingrese un número entero de 4 dígitos : 2345
La suma de los dígitos de 2345 es 14
Presione una tecla para continuar . . .
```

Observaciones:

1) La condición que hemos usado para obligar al usuario a ingresar un número de 4 dígitos es $\text{floor}(1+\log_{10}(\text{abs}(\text{Num})))!=4$ que se vale del hecho que $1+\log_{10}(\text{abs}(\text{Num}))$ entrega un número decimal que está entre $n-1$ y n donde n es el número de dígitos de Num .

Por otro lado, la función $\text{floor}(x)$, perteneciente a la biblioteca `math.h`, permite obtener el entero más grande que no sea mayor que x , de esta forma logramos dejar fuera los decimales.

2) Para obtener los dígitos de Num nos valemos del hecho de que son el resto de dividir Num por $10^0 = 1$ (unidad), por $10^1 = 10$ (decena), por $10^2 = 100$ (centena) y por $10^3 = 1000$ (unidad de mil).

12

Escriba un programa en C que calcule y muestre por pantalla, con 1 decimal, el promedio ponderado de 5 notas de un alumno. El usuario es quien ingresa las notas y las ponderaciones correspondientes como porcentaje (por ejemplo, 20 ó 25).

El programa además debe obligar al usuario a ingresar notas entre 1 y 7, inclusive.

El programa debe mostrar un mensaje indicando si el alumno aprobó (promedio ponderado ≥ 4.0) o si reprobó (promedio ponderado < 4.0)

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    float nota1, nota2, nota3, nota4, nota5, PromPond;
    int p1, p2, p3, p4, p5;

    printf("\nIngrese las 5 notas\n");
    do
    {
        printf("\nNota 1 : ");
        scanf("%f",&nota1);
    }
    while(nota1<1 || nota1>7);

    do
    {
        printf("\nNota 2 : ");
        scanf("%f",&nota2);
    }
    while(nota2<1 || nota2>7);

    do
    {
        printf("\nNota 3 : ");
        scanf("%f",&nota3);
    }
    while(nota3<1 || nota3>7);

    do
    {
        printf("\nNota 4 : ");
        scanf("%f",&nota4);
    }
    while(nota4<1 || nota4>7);
```

```

do
{
printf("\nNota 5 : ");
scanf("%f",&nota5);
}
while(nota5<1 || nota5>7);

printf("\nIngrese las ponderaciones\n");

printf("\nPonderaci%cn 1 : ",162);
scanf("%i",&p1);

printf("\nPonderaci%cn 2 : ",162);
scanf("%i",&p2);

printf("\nPonderaci%cn 3 : ",162);
scanf("%i",&p3);

printf("\nPonderaci%cn 4 : ",162);
scanf("%i",&p4);

printf("\nPonderaci%cn 5 : ",162);
scanf("%i",&p5);

PromPond=(nota1*p1+nota2*p2+nota3*p3+nota4*p4+nota5*p5)/100;

if(PromPond>=3.95)
{
printf("\nEl alumno aprob%c con nota %0.1f\n",162,PromPond);
}
else
{
printf("\nEl alumno reprob%c con nota %0.1f\n",162,PromPond);
}

printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida 1

```
Ingrese las 5 notas
Nota 1 : 3.1
Nota 2 : 2.5
Nota 3 : 5.5
Nota 4 : 3.6
Nota 5 : 5.1
Ingrese las ponderaciones
Ponderación 1 : 26
Ponderación 2 : 14
Ponderación 3 : 20
Ponderación 4 : 20
Ponderación 5 : 20
El alumno aprobó con nota 4.0
Presione una tecla para continuar . . .
```

Salida 2

```
Ingrese las 5 notas
Nota 1 : 6.4
Nota 2 : 6.6
Nota 3 : 1.2
Nota 4 : 1.3
Nota 5 : 4.6
Ingrese las ponderaciones
Ponderación 1 : 35
Ponderación 2 : 25
Ponderación 3 : 10
Ponderación 4 : 15
Ponderación 5 : 15
El alumno aprobó con nota 4.9
Presione una tecla para continuar . . .
```

Observaciones:

- 1) Dado que el usuario ingresa ponderaciones expresadas en porcentaje, se hace necesario dividir por 100 la suma de las notas multiplicadas por sus ponderaciones.
- 2) Este ejercicio asume que el usuario ingresa ponderaciones tales que la suma de ellas es 100.
- 3) Un alumno aprueba si el promedio ponderado es mayor o igual a 3.95, porque estamos considerando un decimal con redondeo.

14

Escriba un programa en C que calcule y muestre por pantalla el cociente de dos números enteros mayores o iguales que -3 y menores o iguales que 15 , ingresados por el usuario.
El programa debe mostrar por pantalla un mensaje indicando si se está dividiendo por cero.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int n1, n2;

    printf("\nIngrese n%cmoros enteros (-3<=x<=15)\n",163);
    do
    {
        printf("\nPrimer n%cmero : ",163);
        scanf("%i",&n1);
    }
    while(n1<-3 || n1>15);
```

```

do
{
printf("\nSegundo número : ",163);
scanf("%i",&n2);
}
while(n2<-3 || n2>15);

if(n2!=0) printf("\n%i / %i = %0.2f\n",n1,n2,(float)n1/n2);
else printf("\nSe está intentando dividir por cero\n",160);

printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida 1

```

Ingrese números enteros <-3<=x<=15>
Primer número : -7
Primer número : 19
Primer número : 5
Segundo número : -4
Segundo número : 8
5 / 8 = 0.63

Presione una tecla para continuar . . .

```

Salida 2

```

Ingrese números enteros <-3<=x<=15>
Primer número : 6
Segundo número : 0
Se está intentando dividir por cero

Presione una tecla para continuar . . .

```

Observación:

Dado que n_1 y n_2 son números enteros es necesario indicar la conversión de tipo (float) para obtener el cociente adecuadamente.

15

Prepare una aplicación en C, para $\alpha = 0, \frac{\pi}{16}, \frac{\pi}{8}, \dots, 2\pi$, que muestre por pantalla una tabla para cada una de las siguientes cónicas

a) La circunferencia

$$\begin{aligned}x &= r \cos(\alpha) \\y &= r \operatorname{sen}(\alpha)\end{aligned}$$

donde $r > 0$ es ingresado por el usuario

b) La elipse

$$\begin{aligned}x &= a \cos(\alpha) \\y &= b \operatorname{sen}(\alpha)\end{aligned}$$

donde $a > b$ y ambas son positivas e ingresadas por el usuario.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
```

```
int main()
{
    float r, x, y, alfa;
    float a,b;

    do
    {
```

```

printf("\nIngrese r : ");
scanf("%f",&r);
}
while(r<=0.0);

printf("\n\tCIRCUNFERENCIA\n");
printf("\t x \t y \n",x,y);
for(alfa=0;alfa<=2*M_PI+0.1;alfa=alfa+M_PI/16)
{
    x=r*cos(alfa) ; y=r*sin(alfa);
    printf("\t %0.2f \t %0.2f \n",x,y);
}
system("PAUSE");

system("cls");

do
{
    do
    {
        printf("\nIngrese a : ");
        scanf("%f",&a);
    }
    while(a<=0);
    do
    {
        printf("\nIngrese b : ");
        scanf("%f",&b);
    }
    while(b<=0);
}
while(a<=b);

printf("\n\tELIPSE\n");
printf("\t x \t y \n",x,y);
for(alfa=0;alfa<=2*M_PI+0.1;alfa=alfa+M_PI/16)
{
    x=a*cos(alfa) ; y=b*sin(alfa);
    printf("\t %0.2f \t %0.2f \n",x,y);
}

```



```
printf("\n\n");
system("PAUSE");
return 0;
}
```

Salida

```
Ingrese r : 1
CIRCUNFERENCIA
  x      y
 1.00   0.00
 0.98   0.20
 0.92   0.38
 0.83   0.56
 0.71   0.71
 0.56   0.83
 0.38   0.92
 0.20   0.98
 0.00   1.00
-0.20   0.98
-0.38   0.92
-0.56   0.83
-0.71   0.71
-0.83   0.56
-0.92   0.38
-0.98   0.20
-1.00  -0.00
-0.98  -0.20
-0.92  -0.38
-0.83  -0.56
-0.71  -0.71
-0.56  -0.83
-0.38  -0.92
-0.20  -0.98
 0.00  -1.00
 0.20  -0.98
 0.38  -0.92
 0.56  -0.83
 0.71  -0.71
 0.83  -0.56
 0.92  -0.38
 0.98  -0.20
 1.00   0.00
Presione una tecla para continuar . . .
```

```
Ingrese a : 3
```

```
Ingrese b : 1
```

```
ELIPSE
```

```
  x      y
 3.00  0.00
 2.94  0.20
 2.77  0.38
 2.49  0.56
 2.12  0.71
 1.67  0.83
 1.15  0.92
 0.59  0.98
 0.00  1.00
-0.59  0.98
-1.15  0.92
-1.67  0.83
-2.12  0.71
-2.49  0.56
-2.77  0.38
-2.94  0.20
-3.00 -0.00
-2.94 -0.20
-2.77 -0.38
-2.49 -0.56
-2.12 -0.71
-1.67 -0.83
-1.15 -0.92
-0.59 -0.98
 0.00 -1.00
 0.59 -0.98
 1.15 -0.92
 1.67 -0.83
 2.12 -0.71
 2.49 -0.56
 2.77 -0.38
 2.94 -0.20
 3.00  0.00
```

```
Presione una tecla para continuar . . .
```

Observaciones:

1) En este ejercicio se ha decidido mostrar por pantalla los resultados correspondientes a la circunferencia primero, y luego los asociados a la elipse; para ello, se utiliza las sentencias `system("PAUSE");` `system("cls");` que nos permiten hacer una pausa para ver los resultados de la circunferencia y luego limpiar la pantalla (`cls:clear screen`) para mostrar los resultados de la elipse.

2) En el bucle `for(alfa=0;alfa<=2*M_PI+0.1;alfa=alfa+M_PI/16)` el valor de `alfa` se mueve hasta `2*M_PI+0.1` porque si llegase hasta `2*M_PI` el último par de valores no aparecería por problemas con los decimales; recordemos que 2π es una representación simbólica de un número que posee una cantidad infinita de decimales, y en programación al considerar un número finito de ellos no alcanzamos a llegar a 2π , por eso se le suma 0.1 para superar el valor 2π

3) En este ejercicio se observan estructuras do ... while anidadas en el siguiente trozo de código:

```
do
{
do
{
printf("\nIngrese a : ");
scanf("%f",&a);
}
while(a<=0);
do
{
printf("\nIngrese b : ");
scanf("%f",&b);
}
while(b<=0);
}
while(a<=b); 
```

16

Escriba un programa en C que calcule la suma de los pares que van desde 0 hasta N , donde N es un número natural par, ingresado por el usuario. Incluya en la suma el 0 y el N .

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
int N, k;
```

```

int suma=0;

do
{
    printf("\nIngrese un n%cmero par : ",163);
    scanf("%i",&N);
}
while(N%2!=0 || N<=0);

for(k=0;k<=N;k=k+2)
    suma=suma+k;

printf("\nLa suma de los pares desde 0 hasta %i es %i\n",N,suma);

printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida

```

Ingrese un número natural par : 0
Ingrese un número natural par : -2
Ingrese un número natural par : 7
Ingrese un número natural par : 12
La suma de los pares desde 0 hasta 12 es 42
Presione una tecla para continuar . . .

```

Observaciones:

1) El valor de N es un número natural par, por tal motivo se incluye el bloque de código:

```

do
{
    printf("\nIngrese un n%cmero par : ",163);
    scanf("%i",&N);
}

```

```
while(N%2!=0 || N<=0);
```

2) En este ejercicio se incorpora un acumulador de suma para calcular la suma de los pares entre 0 y N, incluyendo los extremos.
Los acumuladores de suma tienen la estructura

```
Acumulador=Acumulador+algo
```

que se lee el nuevo valor de Acumulador es igual al valor anterior de Acumulador más algo.

Los acumuladores de suma siempre deben ser inicializados, es decir, se les debe dar un valor inicial; en el ejercicio esto se hace en la declaración

```
int suma=0;
```

Es común que los acumuladores de suma estén incluidos en alguna estructura iterativa que permita ir actualizando el valor del acumulador; en el ejercicio el acumulador está dentro de un bucle for

```
for(k=0;k<=N;k=k+2)  
    suma=suma+k;
```

3) La sentencia

```
for(k=0;k<=N;k=k+2)  
    suma=suma+k;
```

a medida que itera genera los valores que se indican en la siguiente tabla, suponiendo, por ejemplo, que N=10 y sabiendo que suma tiene valor inicial igual a cero:

Iteración	k	suma
1	0	0+0=0
2	2	0+2=2
3	4	2+4=6
4	6	6+6=12
5	8	12+8=20
6	10	20+10=30



Escriba un programa en C que calcule y muestre por pantalla, con 3 decimales, el resultado de

a) $\sum_{i=6}^{60} (-1)^i \cdot i$

b) $\prod_{n=1}^{50} \frac{n}{n+1}$

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    int suma=0; float prod=1.0;
    int i, n;

    for(i=6;i<=60;i++)
        suma=suma+pow(-1,i)*i;

    for(n=1;n<=50;n++)
        prod=prod*(n/(n+1.0));

    printf("\nLa suma es %i\n",suma);
    printf("\nEl producto es %0.3f\n",prod);

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida

```
La suma es 33
El producto es 0.020
Presione una tecla para continuar . . .
```

Observaciones:

1) Cuando se quiere calcular una sumatoria del tipo $\sum_{i=a}^b x(i)$, entonces siempre podemos usar un acumulador de suma dentro de una estructura for, como se indica a continuación

```
for(i=a;i<=b;i++)
    suma=suma+x(i);
```

2) En este ejercicio usamos un acumulador de producto, que como el nombre lo dice nos permite acumular en una variable los productos de alguna expresión.

Para calcular $\prod_{n=1}^{50} \frac{n}{n+1}$ se escribe el trozo de código

```
for(n=1;n<=50;n++)
    prod=prod*(n/(n+1.0));
```

donde la variable real prod se inicializa en 1.0 (float prod=1.0;), porque si la inicializamos en cero entonces todos los productos serían cero.

19

Escriba un programa en C que solicite al usuario el ingreso de dos números enteros distintos, y exhiba en la consola el número total de impares entre los números ingresados incluyéndolos; además el programa debe mostrar por pantalla la suma de los números pares múltiplos de 7 que están entre los números ingresados sin considerarlos.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int num1, num2, mayor, menor, k, NumImp=0, Suma=0;

    printf("\nIngrese dos números enteros distintos\n",163);
    do
    {
        printf("\nPrimer número : ",163);
        scanf("%d",&num1);
        printf("\nSegundo número : ",163);
        scanf("%d",&num2);
    }
    while(num1==num2);

    if(num1>=num2) { mayor=num1; menor=num2; }
    else { mayor=num2; menor=num1; }

    for(k=menor;k<=mayor;k++)
        if(k%2!=0) NumImp=NumImp+1;

    for(k=menor;k<=mayor;k++)
        if(k%2==0 && k%7==0) Suma=Suma+k;

    printf("\nEl número total de impares entre",163);
    printf(" %i y %i, incluye %ndollos, es %i\n",menor,mayor,130,NumImp);

    printf("\nLa suma de los números pares múltiplos de 7" ,163,163);
    printf(" entre %i y %i es %i\n",menor,mayor,Suma);

    printf("\n\n");
    system("PAUSE");
}
```



```
return 0;  
}
```

Salida

```
Ingrese dos números enteros distintos  
Primer número : 5  
Segundo número : 5  
Primer número : 17  
Segundo número : 3  
El número total de impares entre 3 y 17, incluyéndolos, es 8  
La suma de los números pares múltiplos de 7 entre 3 y 17 es 14  
Presione una tecla para continuar . . .
```

Observaciones:

1) En este ejercicio se obtiene el menor y el mayor de los números enteros ingresados por el usuario, para utilizarlos luego en los ciclos for correspondientes para calcular el total de impares primero y luego la suma de los pares múltiplos de 7.

2) Hemos usado la misma variable k para cada ciclo for; esto es posible porque en cada ciclo for se le asigna un valor inicial y final al índice k utilizado.

22

Use C para mostrar por pantalla la suma de los números pares del 2 al 200.

Solución:

Código Fuente

```
#include <stdio.h>  
#include <stdlib.h>
```

```

int main()
{
    int par, SumaPares;

    SumaPares=0;
    for(par=2;par<=200;par=par+2)
        SumaPares=SumaPares+par;

    printf("\nLa suma de los pares del 2 al 200 es %i\n",SumaPares);

    printf("\n\n");
    system("PAUSE");
    return 0;
}

```

Salida

```

La suma de los pares del 2 al 200 es 10100
Presione una tecla para continuar . . .

```

Observaciones:

1) En este ejercicio se inicializa la variable SumaPares luego de haberla declarado, lo que es perfectamente lícito, pero siempre debe ser declarada e inicializada antes de usarla en el programa.

2) Otra forma de resolver el problema anterior es sabiendo que la suma de los pares del 2 al 200 es igual a $\sum_{k=1}^{100} 2k$ cuya codificación es:

```

#include <stdio.h>
#include <stdlib.h>

int main()
{
    int k, SumaPares;

    SumaPares=0;

```

```

for(k=1;k<=100;k++)
    SumaPares=SumaPares+2*k;

printf("\nLa suma de los pares del 2 al 200 es %i\n",SumaPares);

printf("\n\n");
system("PAUSE");
return 0;
}

```

y cuya salida es idéntica a la anterior.

3) Otra manera de resolver el problema dado es usando una estructura do...while o while ... :

```

#include <stdio.h>
#include <stdlib.h>

int main()
{
    int k, SumaPares;

    SumaPares=0; k=2;

    do
    {
        SumaPares=SumaPares+k;
        k=k+2;
    }
    while(k<=200);

    printf("\nLa suma de los pares del 2 al 200 es %i\n",SumaPares);

    printf("\n\n");
    system("PAUSE");
    return 0;
}

```

cuya salida es idéntica a las anteriores.

Escriba un programa en C que calcule y muestre por pantalla la distancia desde un punto P a una recta en el espacio.
El programa debe obligar al usuario a ingresar las coordenadas del punto P en el primer octante.
El usuario además ingresa las coordenadas del punto de la recta y del vector director.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    float p1, p2, p3;
    float x0, y0, z0;
    float a, b, c;
    float sum1, sum2, sum3, Dist;

    printf("\nDistancia de un punto P a una recta L en el espacio\n");

    printf("\n\nCoordenadas de P\n");
    do
    {
        printf("\n p1 : ");
        scanf("%f",&p1);
        printf("\n p2 : ");
        scanf("%f",&p2);
        printf("\n p3 : ");
        scanf("%f",&p3);
    }
    while(p1<0.0 || p2<0.0 || p3<0.0);

    printf("\n\nCoordenadas del punto de la recta\n");
```

```

printf("\n x0 : ");
scanf("%f",&x0);
printf("\n y0 : ");
scanf("%f",&y0);
printf("\n z0 : ");
scanf("%f",&z0);

printf("\n\nCoordenadas del vector director\n");
printf("\n a : ");
scanf("%f",&a);
printf("\n b : ");
scanf("%f",&b);
printf("\n c : ");
scanf("%f",&c);

sum1=pow(b*(z0-p3)-c*(y0-p2),2);
sum2=pow(c*(x0-p1)-a*(z0-p3),2);
sum3=pow(a*(y0-p2)-b*(x0-p1),2);

Dist=sqrt(sum1+sum2+sum3)/sqrt(a*a+b*b+c*c);

printf("\n\nLa distancia es %f\n",Dist);

printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida

```
Distancia de un punto P a una recta L en el espacio
```

```
Coordenadas de P
```

```
p1 : -2
```

```
p2 : 0
```

```
p3 : 2
```

```
p1 : 0
```

```
p2 : 2
```

```
p3 : 1
```

```
Coordenadas del punto de la recta
```

```
x0 : -1
```

```
y0 : 2
```

```
z0 : -1
```

```
Coordenadas del vector director
```

```
a : 3
```

```
b : 2
```

```
c : -2
```

```
La distancia es 2.222876
```

```
Presione una tecla para continuar . . .
```

Observación:

Supongamos que la recta en el espacio tiene ecuación:

$$L: \begin{cases} x = x_0 + at \\ y = y_0 + bt \\ z = z_0 + ct \end{cases}$$

donde $\vec{r} = [a, b, c]$ es el vector director y $A = (x_0, y_0, z_0)$ es un punto cualquiera de la recta.

Para calcular la distancia desde el punto $P = (p_1, p_2, p_3)$ a la recta L mencionada anteriormente se ocupa la fórmula:

$$d(P, L) = \frac{\|\vec{r} \times \overrightarrow{AP}\|}{\|\vec{r}\|}$$

$$\overrightarrow{AP} = [x_0 - p_1, y_0 - p_2, z_0 - p_3]$$

$$\vec{r} \times \overrightarrow{AP} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ a & b & c \\ x_0 - p_1 & y_0 - p_2 & z_0 - p_3 \end{vmatrix} =$$

$$[b(z_0 - p_3) - c(y_0 - p_2), c(x_0 - p_1) - a(z_0 - p_3), a(y_0 - p_2) - b(x_0 - p_1)]$$

$$\|\vec{r} \times \overrightarrow{AP}\| =$$

$$\sqrt{[b(z_0 - p_3) - c(y_0 - p_2)]^2 + [c(x_0 - p_1) - a(z_0 - p_3)]^2 + [a(y_0 - p_2) - b(x_0 - p_1)]^2}$$

$$\|\vec{r}\| = \sqrt{a^2 + b^2 + c^2}$$

Finalmente la distancia desde la recta $L : \begin{cases} x = x_0 + at \\ y = y_0 + bt \\ z = z_0 + ct \end{cases}$ al punto

$P = (p_1, p_2, p_3)$ es

$$d(P, L) = \frac{\|\vec{r} \times \overrightarrow{AP}\|}{\|\vec{r}\|} =$$

$$\frac{\sqrt{[b(z_0 - p_3) - c(y_0 - p_2)]^2 + [c(x_0 - p_1) - a(z_0 - p_3)]^2 + [a(y_0 - p_2) - b(x_0 - p_1)]^2}}{\sqrt{a^2 + b^2 + c^2}} \quad \square$$

27

Escriba un programa en C que solicite al usuario el ingreso de números enteros positivos y muestre por pantalla aquellos que sean pares y la suma de los impares.

Para indicar el fin del proceso el usuario debe ingresar un cero.

Solución:

Código Fuente

```
#include <stdio.h>
```

```

#include <stdlib.h>

int main()
{
    int np, sumalImpares=0;
    do
    {
        printf("\nIngrese un número entero positivo : ",163);
        scanf("%i",&np);
        if(np%2==0) printf("\t\t\t\t\t%i es par",np);
        else sumalImpares=sumalImpares+np;
    }
    while(np!=0);

    printf("\nLa suma de los impares es %i\n",sumalImpares);

    printf("\n\n");
    system("PAUSE");
    return 0;
}

```

Salida

```

Ingrese un número entero positivo : 4
4 es par
Ingrese un número entero positivo : 3
Ingrese un número entero positivo : 2
2 es par
Ingrese un número entero positivo : 1
Ingrese un número entero positivo : 0
0 es par
La suma de los impares es 4
Presione una tecla para continuar . . .

```

Observación:

En este ejercicio los pares se muestran al interior del ciclo do .. while, pues si lo hiciéramos fuera del ciclo se mostraría el último número ingresado si es que éste fuese par.

La suma de los impares debe ir fuera de la estructura do ... while porque se exhibe la suma de todos los impares ingresados.

Escriba una aplicación de consola en C que calcule y muestre por pantalla en color rojo brillante, la suma de los primeros n múltiplos de 7, donde n es un número natural menor o igual a 15 ingresado por el usuario.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <conio2.h>

int main()
{
    int n, suma;
    int k;

    do
    {
        printf("\nIngrese un número natural menor o igual a 15 : ",163);
        scanf("%d",&n);
    }
    while(n<1 || n>15);

    suma=0;

    for(k=1;k<=n;k++)
        suma=suma+7*k;

    textcolor(LIGHTRED);
    printf("\nLa suma de los primeros %i múltiplos de 7 es %d\n\n",n,163,suma);

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida

```
Ingrese un número natural menor o igual a 15 : 19
Ingrese un número natural menor o igual a 15 : -5
Ingrese un número natural menor o igual a 15 : 8
La suma de los primeros 8 múltiplos de 7 es 252

Presione una tecla para continuar . . .
```

Observación:

La suma de los primeros n múltiplos de 7 está dada por $\sum_{k=1}^n 7k$

31

Escriba un programa en C que muestre por pantalla con 4 decimales :

$$a) 50 + \sum_{k=5}^{100} \frac{k+1}{k}$$
$$b) \prod_{k=6}^{17} (k - 5)$$

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    float suma=50.0; int Producto=1;
    int k;
```

```

for(k=5;k<=100;k++)
    suma=suma+(k+1.0)/k;

for(k=6;k<=17;k++)
    Producto=Producto*(k-5);

printf("\nLa suma es %0.4f\n",suma);
printf("\nEl producto es %i\n",Producto);

printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida

```

La suma es 149.1041
El producto es 479001600

Presione una tecla para continuar . . .

```

Observación:

El acumulador de suma en este ejercicio se ha inicializado en 50 porque el enunciado nos pide calcular $50 + \sum_{k=5}^{100} \frac{k+1}{k}$ lo que nos indica que debemos partir con 50.

32

Escriba un programa en C que muestre por pantalla las sumas parciales

$S_n = \sum_{i=1}^n \frac{1}{i}$ donde n es un número entero positivo ingresado por el usuario.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int n, i; float suma=0.0;

    do
    {
        printf("\nIngrese un entero positivo : ");
        scanf("%d",&n);
    }
    while(n<=0);

    for(i=1;i<=n;i++)
    {
        suma=suma+1.0/i;
        printf("\nSuma parcial %i = %f\n",i,suma);
    }

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida

```
Ingrese un entero positivo : -6
Ingrese un entero positivo : 0
Ingrese un entero positivo : 8
Suma parcial 1 = 1.000000
Suma parcial 2 = 1.500000
Suma parcial 3 = 1.833333
Suma parcial 4 = 2.083333
Suma parcial 5 = 2.283334
Suma parcial 6 = 2.450000
Suma parcial 7 = 2.592857
Suma parcial 8 = 2.717857

Presione una tecla para continuar . . .
```

Observación:

En este ejercicio es importante incluir la acumulación de las sumas al interior del ciclo for porque de ese modo se pueden exhibir por pantalla las sumas parciales.

Si mostrásemos la suma fuera del ciclo for, aparecería sólo la suma final.

34

Escriba un programa en C que calcule la suma $\sum_{i=m}^n \frac{n-m+i}{i}$
donde m y n son ingresados por el usuario, con $m \leq n$.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int m,n, i;
    float suma=0.0;

    do
    {
        printf("\nIngrese m : ");
        scanf("%i",&m);
        printf("\nIngrese n : ");
        scanf("%i",&n);
    }
    while(m>n);

    for(i=m;i<=n;i++)
        suma=suma+(float)(n-m+i)/i;

    printf("\nLa suma es %f\n",suma);

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida

```
Ingrese m : 7
Ingrese n : 2
Ingrese m : 2
Ingrese n : 9
La suma es 20.802780
Presione una tecla para continuar . . .
```

Observación:

En este ejercicio es necesario indicar explícitamente que el resultado de $\frac{n-m+i}{i}$ es un número real, porque las variables m , n e i son enteras. \square

36

Use C para solicitar al usuario el ingreso de 21 números enteros positivos y luego mostrar por pantalla la suma de los impares múltiplos de 5. El programa debe obligar al usuario a ingresar números enteros positivos.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int NumPos, suma=0, cuenta=0;
    do
    {
        do
        {
            printf("\nIngrese un número entero positivo : ",163);
            scanf("%i",&NumPos);
        }
        while(NumPos<=0);
        if(NumPos%2!=0 && NumPos%5==0) suma+=NumPos;
        cuenta++;
    }
    while(cuenta<21);
    printf("\nLa suma de los impares múltiplos de 5 es %i\n",163,suma);

    printf("\n\n");
}
```

```
system("PAUSE");  
return 0;  
}
```

Salida

```
Ingrese un número entero positivo : -10  
Ingrese un número entero positivo : 0  
Ingrese un número entero positivo : 10  
Ingrese un número entero positivo : 15  
Ingrese un número entero positivo : 1  
Ingrese un número entero positivo : 1  
Ingrese un número entero positivo : 1  
Ingrese un número entero positivo : 1  
Ingrese un número entero positivo : 5  
Ingrese un número entero positivo : 20  
Ingrese un número entero positivo : 12  
Ingrese un número entero positivo : 15  
Ingrese un número entero positivo : 1  
Ingrese un número entero positivo : 1  
Ingrese un número entero positivo : 1  
Ingrese un número entero positivo : 1  
Ingrese un número entero positivo : 1  
Ingrese un número entero positivo : 1  
Ingrese un número entero positivo : 1  
Ingrese un número entero positivo : 1  
Ingrese un número entero positivo : 1  
Ingrese un número entero positivo : 1  
Ingrese un número entero positivo : 1  
Ingrese un número entero positivo : 1  
Ingrese un número entero positivo : 1  
Ingrese un número entero positivo : 3  
La suma de los impares múltiplos de 5 es 35  
Presione una tecla para continuar . . .
```


Observaciones:

1) En este ejercicio hemos usado dos estructuras do ... while. La estructura exterior sirve para controlar, mediante un contador, el número de veces que debe ser ingresado un número positivo, y la estructura interior es para verificar si el número ingresado es positivo.

2) Hemos usado la notación abreviada `suma+=NumPos;` para el acumulador de suma.

En lenguaje C es equivalente escribir `suma+=NumPos;` que `suma=suma+NumPos;`

3) La sentencia do ... while exterior tiene como condición `cuenta<21` y no `cuenta<=21` porque hemos inicializado la variable `cuenta` en 0 y no en 1. Si la variable `cuenta` la hubiésemos inicializado en 1, entonces la condición sería `cuenta<=21`

4) Si hubiésemos usado una estructura while ... exterior, entonces el programa sería:

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main()
{
    int NumPos, suma=0, cuenta=0;

    while(cuenta<21)
    {
        do
        {
            printf("\nIngrese un número entero positivo : ",163);
            scanf("%i",&NumPos);
        }
        while(NumPos<=0);
        if(NumPos%2!=0 && NumPos%5==0) suma+=NumPos;
        cuenta++;
    }
```

```
printf("\nLa suma de los impares múltiplos de 5 es %i\n",163,suma);
```

```

printf("\n\n");
system("PAUSE");
return 0;
}

```

5) La orden selectiva if debe ir después del do...while interior, porque nos interesa considerar en la suma sólo los números positivos que cumplan la condición solicitada.

37

Escriba un programa en C que solicite al usuario el ingreso de un número real y luego evalúe en tal número la función

$$f(x) = \begin{cases} \sum_{i=1}^{50} \frac{1}{x^i} & , \text{ si } x > \sqrt{3} \\ \prod_{k=2}^6 (x - k) & , \text{ si } x \leq \sqrt{3} \end{cases}$$

mostrando por pantalla, con 2 decimales, la función evaluada en el punto ingresado por el usuario.

Solución:

Código Fuente

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    float x, suma, producto, fx;
    int i, k;

    suma=0.0; producto=1.0;
    printf("\nIngrese un número real : ",163);
    scanf("%f",&x);

```

```

if(x>sqrt(3))
{
for(i=1;i<=50;i++)
suma=suma+1.0/pow(x,i);
fx=suma;
}
else
{
for(k=2;k<=6;k++)
producto=producto*(x-k);
fx=producto;
}
printf("\nf(%0.2f) = %0.2f\n",x,fx);

printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida 1

```

Ingrese un número real : 4
f<4.00> = 0.33
Presione una tecla para continuar . . .

```

```

Ingrese un número real : 1.7
f<1.70> = -12.73
Presione una tecla para continuar . . .

```

Observaciones:

- 1) En este ejercicio exhibimos el resultado al final del if porque almacenamos tales resultados en la variable fx
- 2) En este ejercicio no es necesario el uso de dos variables índice distintas (i y k), se hace sólo para relacionar el enunciado con el código fuente.

38

Use C para determinar para qué valor de n la productoria $\prod_{z=1}^n \frac{z+5}{z+1}$ es mayor o igual a 2013.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int n=0, z; float prod;
    do
    {
        prod=1.0;
        n++;
        for(z=1;z<=n;z++)
            prod=prod*((z+5.0)/(z+1.0));
    }
    while(prod<2013.0);

    printf("\nEl valor de n de modo que la productoria sea mayor o igual que 2013 es %i\n",n);
    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida

```
El valor de n de modo que la productoria sea mayor o igual que 2013 es 19
Presione una tecla para continuar . . .
```

Observaciones:

1) Las líneas

```
for(z=1;z<=n;z++)  
    prod=prod*((z+5.0)/(z+1.0));
```

permiten calcular la productoria $\prod_{z=1}^n \frac{z+5}{z+1}$

2) A medida que cambia el valor de n debemos nuevamente inicializar prod en 1 para que no se acumulen las productorias previas.

GUÍA DE PRÁCTICO N° 8

NÚMEROS SEUDOALEATORIOS

Teoría Números Seudoaleatorios

33 **Números Seudoaleatorios**

Algunos programas requieren la obtención de números al azar para, por ejemplo, simular el lanzamiento de un dado.

En C es posible generar números pseudoaleatorios usando la función `rand()` perteneciente a la biblioteca `stdlib.h`, con la que se obtiene un número pseudoaleatorio entero en el intervalo `[0, RAND_MAX]`. La constante `RAND_MAX` en DevC++ es igual a 32767.

34 **Semilla Aleatoria**

La generación de números pseudoaleatorios requiere que cada vez que se ejecute un programa, la sucesión de números pseudoaleatorios cambie. Para ello existe la función `srand()` perteneciente a la librería `stdlib.h`, que permite la generación de una semilla aleatoria dependiente de algún argumento.

La sentencia `srand(time(NULL));` permite crear una semilla que depende del tiempo transcurrido, en segundos, desde el 1 de enero de 1970 hasta el instante en que se ejecuta el programa.

La sentencia `srand(time(NULL));` debe aparecer sólo una vez en el programa y debe ser antes del uso de la función `rand()`. Si la función `rand()` es usada sin o antes de `srand()`, la secuencia de números pseudoaleatorios siempre será la misma en cada ejecución del programa.

La sentencia `srand(time(NULL));` nos obliga a incluir la biblioteca `time.h` al inicio del programa, porque la función `time` pertenece a tal biblioteca.

T 35 **Números Pseudoaleatorios en un Intervalo Arbitrario**

En DevC++ la función `rand()` genera números enteros pseudoaleatorios en el intervalo $[0, 32767]$, pero generalmente los números que necesitamos no pertenecen a tal intervalo, o quizás no sean enteros.

Un camino para obtener números pseudoaleatorios en un intervalo arbitrario es el uso del operador módulo `%` que nos entrega el resto de dividir dos números enteros.

Por ejemplo, la orden `rand()%7` nos entregará un valor entero que va de 0 a 6, porque el resto de dividir un número entero por 7 puede ser 0, 1, 2, ..., 6.

Otro ejemplo puede ser, obtener números pseudoaleatorios enteros en el intervalo $[-2, 5]$, para ello podemos escribir `rand()%8-2` porque `rand()%8` entrega números que van de 0 a 7, y si a estos números le restamos 2, los tales irán de -2 a 5.

En general, para obtener un número entero pseudoaleatorio en el intervalo $[0, N]$ podemos escribir `rand()%(N+1)`

Ahora, para obtener un número entero pseudoaleatorio en el intervalo $[M, N]$ podemos escribir `rand()%(N-M+1)+M`

Por otro lado, si queremos números pseudoaleatorios con 1 decimal podemos amplificar el intervalo por 10, y luego el resultado dividirlo por 10. Por ejemplo, si queremos números del 1 al 7 con 1 decimal, podemos escribir `(rand()%61+10)/10.0`

Además, si queremos números pseudoaleatorios con 2 decimales, podemos amplificar el intervalo por 100, y luego el resultado dividirlo por 100. Por ejemplo, si queremos números del 1 al 7 con 2 decimales, podemos escribir `(rand()%601+100)/100.0`

Ejercicios

Números Pseudoaleatorios

[1] Escriba un programa en C que calcule y muestre por pantalla la suma y el producto de dos números aleatorios enteros generados por el computador.

[2] Use C para mostrar por pantalla la suma, la resta y el producto de dos números aleatorios enteros, mayores que 1 y menores que 45.

[3] Escriba un programa en C que permita calcular el promedio de 3 notas, del 1 al 7, generadas al azar.

[4] Escriba un programa en C que calcule y muestre por pantalla la suma de 100 números aleatorios enteros positivos no mayores que 19, además de los números aleatorios generados.

[5] Escriba un programa en C que genere aleatoriamente un número real x con un decimal en el intervalo $[-1, 6]$, y luego evalúe y muestre por pantalla el resultado $f(x)$, donde

$$f(x) = \begin{cases} 3^{x+3} & , \text{ si } -1 \leq x < 0 \\ \text{sen}(x) & , \text{ si } 0 \leq x < \frac{\pi}{2} \\ \sqrt[3]{x + |x|} & , \text{ si } \frac{\pi}{2} \leq x \leq 6 \end{cases}$$

[6] Escriba un programa en C que solicite al usuario el ingreso de los resultados de la tabla del N, donde N es un número aleatorio del 1 al 12. El programa debe mostrar por pantalla el número de aciertos y el resultado correcto en caso de error.

[7] Escriba un programa en C que permita al usuario "adivinar" cuál es el número del 1 al 21 que "pensó" el computador, mostrando por pantalla el número total de intentos hasta acertar.

[8] Escriba un programa en C que simule el lanzamiento de un dado cien veces, mostrando al final del proceso el porcentaje de veces que salió el 1, el porcentaje asociado al 2, y así sucesivamente.

[9] Escriba un programa en C que muestre por pantalla la suma de todos los pares múltiplos de 6 y el producto de todos los impares múltiplos de 9, de un conjunto de 50 números aleatorios generados por el computador, entre -7 y 101 inclusive.

[10] Escriba un programa en C que permita al usuario "adivinar" cuál es el número del 1 al 10 que "pensó" el computador. El programa debe mostrar por pantalla el número de intentos hasta "adivinar" el número, y mostrar un mensaje apropiado, en el color indicado, de acuerdo a la siguiente tabla:

N° intentos	Mensaje	Color
1	Extraordinario	Azul
2	Excelente	Rojo Brillante
3-5	Bien	Verde
> 5	Mal	Magenta

[11] Genere aleatoriamente el tiempo, en minutos, de 20 llamadas de larga distancia internacional país frecuente, y luego calcule el costo asociado a cada una de ellas si se sabe que hasta 10 minutos el valor es \$950 y por cada minuto adicional, todo horario, es \$100. Suponga que el tiempo para cada llamada no puede superar los 300 minutos.

[12] Escriba una aplicación de consola que muestre por pantalla 10 números aleatorios pares positivos no mayores que 110.

[13] Usando los valores del código ASCII extendido, muestre por pantalla, separados por un tabulación, 50 caracteres de tal código generados aleatoriamente.

[14] Escriba un programa en C que muestre por pantalla cuáles son los múltiplos de 5 que están entre 1 y N , donde N es un número entero aleatorio no mayor que 1000.

[15] Escriba un programa en C que calcule el producto de dos números enteros mayores o iguales que 1 y menores o iguales que 57 generados aleatoriamente. Si tal producto es mayor o igual que 1000 debe calcular además el cociente de los números y si es menor que 1000 debe calcular además la suma.

[16] Escriba un programa en C que genere aleatoriamente 500 números enteros en el intervalo del -235 al 578 inclusive, y muestre por pantalla la suma de los múltiplos de 3 y de 9. Además el programa debe mostrar por pantalla el total de números impares.

[17] Escriba un programa en C que permita lanzar una moneda las veces que indique el usuario y luego muestre por pantalla el número de veces que apareció cara y el número de veces que apareció sello.

[18] Escriba un programa en C que genere aleatoriamente tres números enteros positivos no mayores que 50, y determine el mayor de ellos.

[19] Escriba un programa en C que solicite al usuario que adivine cuál es el número entero del 1 al 17 que generó el computador. Se deben mostrar los siguientes mensajes de acuerdo al número total de intentos:

Intentos	Mensaje
1, 2 ó 3	Muy bien
4, 5 ó 6	Bien
7, 8 ó 9	Mal
>9	Muy Mal

[20] Escriba un programa en C que simule el lanzamiento de un dado mil veces, mostrando al final del proceso el porcentaje de veces que salió el 1, el porcentaje asociado al 2, y así sucesivamente. El programa además debe mostrar un mensaje, en rojo brillante, indicando cuál fue la cara que apareció más veces.

[21] Escriba un programa en C que permita calcular el promedio de n notas, del 1 al 7, generadas al azar, donde n es un número natural ingresado por el usuario. Exhiba las notas en la consola separadas por una tabulación.

[22] Escriba una aplicación de consola en C que permita al usuario ejercitar con las tablas de multiplicar del 1 al 12. Los ejercicios se repiten hasta que el usuario termina de ejercitar cuando ingresa como resultado de la multiplicación un -1 . El programa debe mostrar al final del proceso el porcentaje de aciertos. Los números a multiplicar deben ser generados aleatoriamente.

PROBLEMAS RESUELTOS

NÚMEROS SEUDOALEATORIOS

1

Escriba un programa en C que calcule y muestre por pantalla la suma y el producto de dos números aleatorios enteros generados por el computador.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    int num1, num2;

    srand(time(NULL));

    num1=rand();
    num2=rand();

    printf("\n\t%i + %i = %i\n",num1,num2,num1+num2);

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida

```
27410 + 24479 = 51889
```

```
Presione una tecla para continuar . . .
```

Observaciones:

1) Para trabajar con números aleatorios es necesario incluir dos cosas en nuestro programa:

a) La inicialización del generador de números aleatorios mediante la sentencia `srand(time(NULL))`; que permite generar una semilla (seed) aleatoria (`rand`), que en este caso la hacemos dependiente del instante (`time`) en que se ejecuta el programa.

La función `time` está incluida en la biblioteca `time.h` y en estricto rigor deberíamos incluirla en la sección de los `include`, pero el entorno DevC++ nos permite usarla sin incluir tal librería; de todos modos, por un tema de compatibilidad con otros entornos, es recomendable incluir la biblioteca `time.h`, cuando se requiera.

b) La generación de números aleatorios mediante la función `rand()` entrega como resultado un número aleatorio entero en el intervalo `[0,RAND_MAX]`, donde `RAND_MAX` es, a lo menos, 32767 (esto depende del compilador que usemos).

2) A veces se tiende a codificar el problema anterior de la siguiente forma:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    srand(time(NULL));

    printf("\n%i + %i = %i\n",rand(),rand(),rand()+rand());

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

```
}
```

cuya salida es:

```
5683 + 3031 = 58003
Presione una tecla para continuar . . .
```

El defecto del programa anterior es que `rand()+rand()` es la suma de números pseudoaleatorios que no necesariamente se corresponden con los primeros mencionados, es decir, con los sumandos.

3) Si en la semilla aleatoria se considera un argumento constante, como por ejemplo `srand(5)`, entonces los números aleatorios siempre serán los mismos en cada ejecución de la aplicación.

Así, el código fuente:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    int num1, num2;

    srand(5);

    num1=rand();
    num2=rand();

    printf("\n\t%i + %i = %i\n",num1,num2,num1+num2);

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

tiene, después de cada ejecución, la salida única:

```
54 + 28693 = 28747
```

```
Presione una tecla para continuar . . .
```



2

Use C para mostrar por pantalla la suma, la resta y el producto de dos números aleatorios enteros, mayores que 1 y menores que 45.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    int num1, num2;

    srand(time(NULL));

    num1=rand()%43+2;
    num2=rand()%43+2;

    printf("\n%i + %i = %i\n",num1,num2,num1+num2);
    printf("\n%i - %i = %i\n",num1,num2,num1-num2);
    printf("\n%i * %i = %i\n",num1,num2,num1*num2);

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida

```
4 + 23 = 27
4 - 23 = -19
4 * 23 = 92

Presione una tecla para continuar . . .
```

Observaciones:

1) Cada vez que queramos generar números aleatorios que pertenezcan a un cierto intervalo, podemos hacer uso del operador módulo. En efecto, $X\%N$ genera un número entero perteneciente al intervalo $[0, N-1]$, lo que nos permite generar números aleatorios en un cierto intervalo.

En el ejercicio debemos obtener números aleatorios en el intervalo cerrado $[2, 44]$, porque el enunciado dice "números aleatorios enteros, mayores que 1 y menores que 45."; para ello $\text{rand}\()\%43$ permitirá generar un número aleatorio en $[0, 42]$ y si a ese intervalo le sumamos 2 obtenemos el intervalo $[2, 44]$ que es el que necesitamos. La sentencia correspondiente, por tanto, para generar un número aleatorio mayor que 1 y menor que 45 es $\text{rand}\()\%43+2$

2) Escribir $\text{rand}\()\%43+2$ es equivalente a escribir $(\text{rand}\()\%43) + 2$, es decir, primero se calcula el módulo y luego al resultado, se le suma 2

3) Para generar números aleatorios en un intervalo $[M, N]$, donde M y N son números enteros, se usa la expresión $\text{rand}\()\%(N+1-M)+M$

4

Escriba un programa en C que calcule y muestre por pantalla la suma de 100 números aleatorios enteros positivos no mayores que 19, además de los números aleatorios generados.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    srand(time(NULL));

    int suma=0, i, x;

    for(i=1;i<=100;i++)
    {
        x=rand()%19+1; printf("\t%i\t",x);
        suma+=x;
    }
    printf("\nLa suma es %i\n",suma);

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida

```
9      13      1      12      1
4      18      3      15      11
3      8       7      17      10
18     16     4      18      12
11     19     12     13      11
10     16     10     16      5
2      4      10     14      8
9      13     17     12      16
14     8      19     19      7
6      12     14     5       13
2      8      13     15      6
12     6      12     16      7
15     19     15     3       1
14     9      15     12      12
17     17     19     10     19
4      18     8      10     4
16     8      17     9      19
15     3      13     2      4
8      5      1      10     4
3      5      6      2      10
```

La suma es 1043

Presione una tecla para continuar . . .

Observaciones:

- 1) Un número entero positivo no mayor que 19 es un número entero que pertenece al intervalo cerrado $[1,19]$
- 2) En este ejercicio, en vez de usar un ciclo for podríamos haber usado una estructura while... o do...while con el respectivo contador, para llevar el control de repeticiones.

5

Escriba un programa en C que genere aleatoriamente un número real x con un decimal en el intervalo $[-1,6]$, y luego evalúe y muestre por pantalla el resultado $f(x)$, donde

$$f(x) = \begin{cases} 3^{x+3} & , \text{ si } -1 \leq x < 0 \\ \text{sen}(x) & , \text{ si } 0 \leq x < \frac{\pi}{2} \\ \sqrt[3]{x + |x|} & , \text{ si } \frac{\pi}{2} \leq x \leq 6 \end{cases}$$

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>

int main()
{
    float x, fx;
    srand(time(NULL));

    x=(rand()%71-10)/10.;

    if(x>=-1.0 && x<0.0) fx=pow(3,x+3);
    else if(x>=0.0 && x<M_PI/2) fx=sin(x);
    else if(x>=M_PI/2 && x<=6.0) fx=pow(x+fabs(x),1./3);

    printf("\nf(%0.1f) = %0.1f\n",x,fx);

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida 1

```
f<2.4> = 1.7
Presione una tecla para continuar . . .
```

Salida 2

```
f<-0.2> = 21.7
Presione una tecla para continuar . . .
```

Salida 3

```
f<1.0> = 0.8
```

```
Presione una tecla para continuar . . .
```

Observación:

En este ejercicio debemos generar números aleatorios, con un decimal, que estén en el intervalo $[-1,6]$; para ello se obtienen números en el intervalo $[-10,60]$ los que se dividen por 10, para finalmente lograr números con un decimal en el intervalo especificado.

6

Escriba un programa en C que solicite al usuario el ingreso de los resultados de la tabla del N, donde N es un número aleatorio del 1 al 12. El programa debe mostrar por pantalla el número de aciertos y el resultado correcto en caso de error.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    srand(time(NULL));

    int N, i, Res, aciertos=0;

    N=rand()%12+1;
```

```

for(i=1;i<=12;i++)
{
    printf("\n%i * %i = ",i,N);
    scanf("%i",&Res);
    if(Res==i*N) aciertos++;
    else printf("\nEl resultado correcto es %i\n",i*N);
}
printf("\nEl total de aciertos es %i\n",aciertos);

printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida

```

1 * 3 = 3
2 * 3 = 5
El resultado correcto es 6
3 * 3 = 7
El resultado correcto es 9
4 * 3 = 12
5 * 3 = 15
6 * 3 = 18
7 * 3 = 21
8 * 3 = 22
El resultado correcto es 24
9 * 3 = 27
10 * 3 = 30
11 * 3 = 33
12 * 3 = 45
El resultado correcto es 36
El total de aciertos es 8

Presione una tecla para continuar . . .

```



7

Escriba un programa en C que permita al usuario "adivinar" cuál es el número del 1 al 21 que "pensó" el computador, mostrando por pantalla el número total de intentos hasta acertar.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    int NumComp, NumUsuario, intentos=1;

    srand(time(NULL));

    NumComp=rand()%21+1;

    do
    {
        printf("\nIngrese el número que pensó el computador : ",163,162);
        scanf("%i",&NumUsuario);
        if(NumComp!=NumUsuario)
        {
            printf("\nNo es el número que pensó\n",163,130);
            intentos++;
        }
    }
    while(NumComp!=NumUsuario);

    printf("\nAcertaste en %i intentos",intentos);

    printf("\n\n");
    system("PAUSE");
}
```

```
    return 0;  
}
```

Salida

```
Ingrese el número que pensó el computador : 11  
No es el número que pensé  
Ingrese el número que pensó el computador : 2  
No es el número que pensé  
Ingrese el número que pensó el computador : 6  
No es el número que pensé  
Ingrese el número que pensó el computador : 16  
No es el número que pensé  
Ingrese el número que pensó el computador : 5  
No es el número que pensé  
Ingrese el número que pensó el computador : 12  
Acertaste en 6 intentos  
Presione una tecla para continuar . . .
```

Observación:

La sentencia `NumComp=rand()%21+1;` debe ir fuera del ciclo `do...while` porque si estuviese al interior, entonces en cada nuevo intento se generaría un número aleatorio distinto.

8

Escriba un programa en C que simule el lanzamiento de un dado cien veces, mostrando al final del proceso el porcentaje de veces que salió el 1, el porcentaje asociado al 2, y así sucesivamente.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    int NumDado, i;
    int cuenta1=0, cuenta2=0, cuenta3=0, cuenta4=0, cuenta5=0, cuenta6=0;

    srand(time(NULL));

    for(i=1;i<=100;i++)
    {
        NumDado=rand()%6+1;
        if(NumDado==1) cuenta1++;
        else if(NumDado==2) cuenta2++;
        else if(NumDado==3) cuenta3++;
        else if(NumDado==4) cuenta4++;
        else if(NumDado==5) cuenta5++;
        else cuenta6++;
    }
    printf("\nEl porcentaje de veces que sali%c el 1 es %i%%\n",162,cuenta1);
    printf("\nEl porcentaje de veces que sali%c el 2 es %i%%\n",162,cuenta2);
    printf("\nEl porcentaje de veces que sali%c el 3 es %i%%\n",162,cuenta3);
    printf("\nEl porcentaje de veces que sali%c el 4 es %i%%\n",162,cuenta4);
    printf("\nEl porcentaje de veces que sali%c el 5 es %i%%\n",162,cuenta5);
    printf("\nEl porcentaje de veces que sali%c el 6 es %i%%\n",162,cuenta6);

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```


Salida

```
El porcentaje de veces que salió el 1 es 19%
El porcentaje de veces que salió el 2 es 19%
El porcentaje de veces que salió el 3 es 13%
El porcentaje de veces que salió el 4 es 15%
El porcentaje de veces que salió el 5 es 17%
El porcentaje de veces que salió el 6 es 17%

Presione una tecla para continuar . . .
```

Observaciones:

1) Si el número de lanzamientos no hubiese sido 100, sino por ejemplo 550, entonces el código fuente podría ser:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    int NumDado, i, NumLanzam=550;
    int cuenta1=0, cuenta2=0, cuenta3=0, cuenta4=0, cuenta5=0, cuenta6=0;

    srand(time(NULL));

    for(i=1;i<=NumLanzam;i++)
    {
        NumDado=rand()%6+1;
        if(NumDado==1) cuenta1++;
        else if(NumDado==2) cuenta2++;
        else if(NumDado==3) cuenta3++;
        else if(NumDado==4) cuenta4++;
        else if(NumDado==5) cuenta5++;
        else cuenta6++;
    }
    printf("\nEl porcentaje de veces que sali%c el 1 es %0.2f%%\n",162,100.0*cuenta1/NumLanzam);
    printf("\nEl porcentaje de veces que sali%c el 2 es %0.2f%%\n",162,100.0*cuenta2/NumLanzam);
    printf("\nEl porcentaje de veces que sali%c el 3 es %0.2f%%\n",162,100.0*cuenta3/NumLanzam);
    printf("\nEl porcentaje de veces que sali%c el 4 es %0.2f%%\n",162,100.0*cuenta4/NumLanzam);
    printf("\nEl porcentaje de veces que sali%c el 5 es %0.2f%%\n",162,100.0*cuenta5/NumLanzam);
    printf("\nEl porcentaje de veces que sali%c el 6 es %0.2f%%\n",162,100.0*cuenta6/NumLanzam);
```

```
printf("\n\n");
system("PAUSE");
return 0;
}
```

cuya salida es:

```
El porcentaje de veces que salió el 1 es 13.45%
El porcentaje de veces que salió el 2 es 19.27%
El porcentaje de veces que salió el 3 es 17.45%
El porcentaje de veces que salió el 4 es 16.55%
El porcentaje de veces que salió el 5 es 17.27%
El porcentaje de veces que salió el 6 es 16.00%

Presione una tecla para continuar . . .
```

2) Para lograr que aparezca el signo de porcentaje se debe anotar %%

3) Si no se sabe cuál es el total de lanzamientos, entonces lo recomendable es usar while... o do...while, junto con un contador que permita llevar el registro de las veces que se ha lanzado el dado.

9

Escriba un programa en C que muestre por pantalla la suma de todos los pares múltiplos de 6 y el producto de todos los impares múltiplos de 9, de un conjunto de 50 números aleatorios generados por el computador, entre -7 y 101 inclusive.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
```

```

int main()
{
    int total=50, k, num;
    int sum=0, prod=1;

    srand(time(NULL));

    for(k=1;k<=total;k++)
    {
        num=rand()%109-7;
        if(num%2==0 && num%6==0) sum+=num;
        if(num%2!=0 && num%9==0) prod*=num;
    }

    printf("\nLa suma de los pares m%cltiplos de 6 es %i\n",163,sum);
    printf("\nEl producto de los impares m%cltiplos de 9 es %i\n",163,prod);

    printf("\n\n");
    system("PAUSE");
    return 0;
}

```

Salida

```

La suma de los pares múltiplos de 6 es 354
El producto de los impares múltiplos de 9 es 45927

Presione una tecla para continuar . . .

```

Observaciones:

- 1) Existe también una forma abreviada para el acumulador de producto `prod=prod*num;` que es `prod*=num;`
- 2) Un número es impar, si el resto de dividirlo por 2 es distinto de cero o igual a 1.

11

Genere aleatoriamente el tiempo, en minutos, de 20 llamadas de larga distancia internacional país frecuente, y luego calcule el costo asociado a cada una de ellas si se sabe que hasta 10 minutos el valor es \$950 y por cada minuto adicional, todo horario, es \$100.
Suponga que el tiempo para cada llamada no puede superar los 300 minutos.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    srand(time(NULL));

    int k, tiempo, costo;

    for(k=1;k<=20;k++)
    {
        tiempo=rand()%300+1;
        if(tiempo<=10) costo=950;
        else costo=950+100*tiempo;
        printf("\nLa llamada %i fue de %i minutos y cost%c %i pesos\n",k,tiempo,162,costo);
    }

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida

```
La llamada 1 fue de 46 minutos y costó 5550 pesos
La llamada 2 fue de 107 minutos y costó 11650 pesos
La llamada 3 fue de 238 minutos y costó 24750 pesos
La llamada 4 fue de 69 minutos y costó 7850 pesos
La llamada 5 fue de 220 minutos y costó 22950 pesos
La llamada 6 fue de 220 minutos y costó 22950 pesos
La llamada 7 fue de 119 minutos y costó 12850 pesos
La llamada 8 fue de 130 minutos y costó 13950 pesos
La llamada 9 fue de 5 minutos y costó 950 pesos
La llamada 10 fue de 291 minutos y costó 30050 pesos
La llamada 11 fue de 225 minutos y costó 23450 pesos
La llamada 12 fue de 2 minutos y costó 950 pesos
La llamada 13 fue de 185 minutos y costó 19450 pesos
La llamada 14 fue de 28 minutos y costó 3750 pesos
La llamada 15 fue de 53 minutos y costó 6250 pesos
La llamada 16 fue de 221 minutos y costó 23050 pesos
La llamada 17 fue de 280 minutos y costó 28950 pesos
La llamada 18 fue de 199 minutos y costó 20850 pesos
La llamada 19 fue de 65 minutos y costó 7450 pesos
La llamada 20 fue de 196 minutos y costó 20550 pesos

Presione una tecla para continuar . . .
```

Observación:

La función que permite calcular el costo asociado a cada llamada es:

$$\text{costo}(\text{tiempo}) = \begin{cases} 950 & , 1 \leq \text{tiempo} \leq 10 \\ 950 + 100 * \text{tiempo} & , 10 < \text{tiempo} \leq 300 \end{cases} \quad \square$$

12

Escriba una aplicación de consola que muestre por pantalla 10 números aleatorios pares positivos no mayores que 110.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    int Num, contador=0;
    srand(time(NULL));

    do
    {
        Num=rand()%109+2;
        if(Num%2==0)
        {
            contador++;
            printf("\nPar numero %i : %i\n",163,contador,Num);
        }
    }
    while(contador<10);

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida

```
Par número 1 : 104
Par número 2 : 42
Par número 3 : 66
Par número 4 : 58
Par número 5 : 82
Par número 6 : 64
Par número 7 : 30
Par número 8 : 108
Par número 9 : 100
Par número 10 : 82

Presione una tecla para continuar . . .
```

Observaciones:

- 1) En este ejercicio hemos ocupado una estructura do..while porque no sabemos de antemano si los números aleatorios generados van a ser pares o no.
- 2) Los pares positivos no mayores que 110, son números que se mueven en el intervalo del 2 al 110, incluyendo los extremos.
- 3) La variable contador sirve para contar los pares que se han obtenido, número, que de acuerdo al enunciado, no debe superar a 10.

13

Usando los valores del código ASCII extendido, muestre por pantalla, separados por un tabulación, 50 caracteres de tal código generados aleatoriamente.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
```

```
#include <time.h>

int main()
{
    srand(time(NULL));

    int k;

    for(k=1;k<=50;k++)
        printf("\t%i %c",k,rand()%128+128);

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida

```

10 é    11 ÷    12 μ    13 `f    14 ç    15 ó    16 ¨    17 ■    18 ı    19 ¶
20 ä    21 ■    22 μ    23 ã    24 º    25 ó    26 ÷    27 Å    28 ı    29 ı
30 ū    31 -    32 0    33 ð    34 ı    35 ¸    36 -    37 ¸    38 =    39 ú    é
40 °    41 î    42 º    43 .    44 ¶    45 ı    46 «    47 ĩ    48 ¶    49 é
50 È

Presione una tecla para continuar . . .
```

Observación:

Recordemos que el código ASCII extendido considera valores del 128 al 255.

17

Escriba un programa en C que permita lanzar una moneda las veces que indique el usuario y luego muestre por pantalla el número de veces que apareció cara y el número de veces que apareció sello.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    int NumLanzam, i, Moneda, CuentaCara=0, CuentaSello=0;

    srand(time(NULL));

    printf("\nIndique el número de veces que quiere lanzar una moneda : ",163);
    scanf("%i",&NumLanzam);

    for(i=1;i<=NumLanzam;i++)
    {
        Moneda=rand()%2+1; //1=cara  2=sello
        if(Moneda==1) CuentaCara++;
        else CuentaSello++;
    }

    printf("\nCara apareció %i veces y sello %i veces\n",162,CuentaCara,CuentaSello);

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida

```
Indique el número de veces que quiere lanzar una moneda : 11
Cara apareció 6 veces y sello 5 veces

Presione una tecla para continuar . . .
```

Observaciones:

1) Para resolver este ejercicio se establece la relación cara=1 y sello=2. Es conveniente elegir dos números enteros consecutivos para que sea fácil la generación aleatoria de aquellos.

2) Es importante darle un valor inicial a las variables CuentaCara y CuentaSello, porque de otro modo podrían almacenar un valor "basura" y los resultados serían impredecibles.

La salida es

```
Indique el número de veces que quiere lanzar una moneda : 11
Cara apareció 73 veces y sello 2697323 veces

Presione una tecla para continuar . . .
```

si se cambia la línea `int NumLanzam, i, Moneda, CuentaCara=0, CuentaSello=0;`
por `int NumLanzam, i, Moneda, CuentaCara, CuentaSello;`

19

Escriba un programa en C que solicite al usuario que adivine cuál es el número entero del 1 al 17 que generó el computador.

Se deben mostrar los siguientes mensajes de acuerdo al número total de intentos:

Intentos	Mensaje

1, 2 ó 3	Muy bien
4, 5 ó 6	Bien
7, 8 ó 9	Mal
>9	Muy Mal

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    int NComp, Num, Intentos=0;
    srand(time(NULL));

    NComp=rand()%17+1;

    do
    {
        printf("\nAdivina el número, del 1 al 17, que pensé : ",163,130);
        scanf("%i",&Num);
        Intentos++;
    }
    while(Num!=NComp);

    if(Intentos==1 || Intentos==2 || Intentos==3) printf("\nMuy bien\n");
    else if(Intentos==4 || Intentos==5 || Intentos==6) printf("\nBien\n");
    else if(Intentos==7 || Intentos==8 || Intentos==9) printf("\nMal\n");
    else printf("\nMuy mal\n");

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida 1

```
Adivina el número, del 1 al 17, que pensé : 5
Adivina el número, del 1 al 17, que pensé : 11
Muy bien
Presione una tecla para continuar . . .
```

Salida 2

```
Adivina el número, del 1 al 17, que pensé : 12
Adivina el número, del 1 al 17, que pensé : 4
Adivina el número, del 1 al 17, que pensé : 16
Adivina el número, del 1 al 17, que pensé : 8
Adivina el número, del 1 al 17, que pensé : 1
Adivina el número, del 1 al 17, que pensé : 3
Bien
Presione una tecla para continuar . . .
```

Salida 3

```
Adivina el número, del 1 al 17, que pensé : 1
Adivina el número, del 1 al 17, que pensé : 2
Adivina el número, del 1 al 17, que pensé : 3
Adivina el número, del 1 al 17, que pensé : 4
Adivina el número, del 1 al 17, que pensé : 5
Adivina el número, del 1 al 17, que pensé : 6
Adivina el número, del 1 al 17, que pensé : 7
Adivina el número, del 1 al 17, que pensé : 8
Adivina el número, del 1 al 17, que pensé : 9
Adivina el número, del 1 al 17, que pensé : 10
Adivina el número, del 1 al 17, que pensé : 11
Adivina el número, del 1 al 17, que pensé : 12
Adivina el número, del 1 al 17, que pensé : 13
Muy mal
Presione una tecla para continuar . . .
```

Observación:

El número de intentos es un valor natural, por tal motivo, se han especificado cada una de las cifras correspondientes en las condiciones que aparecen en los if.

21

Escriba un programa en C que permita calcular el promedio de n notas, del 1 al 7, generadas al azar, donde n es un número natural ingresado por el usuario. Exhiba las notas en la consola separadas por una tabulación.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    int n;
    printf("\nIngrese el total de notas : ");
    scanf("%i",&n);

    srand(time(NULL));

    int i, nota, suma=0;
    for(i=1;i<=n;i++)
    {
        nota=rand()%7+1;
        printf("Nota %i : %i\t",i,nota);
        suma+=nota;
    }

    float promedio;
    promedio=(float) suma/n;

    printf("\n\nEl promedio es : %0.1f\n",promedio);

    printf("\n\n");
    system("PAUSE");
}
```

```
return 0;  
}
```

Salida

```
Ingrese el total de notas : 12  
Nota 1 : 4      Nota 2 : 6      Nota 3 : 3      Nota 4 : 4      Nota 5 : 6  
Nota 6 : 4      Nota 7 : 2      Nota 8 : 5      Nota 9 : 3      Nota 10 : 4  
Nota 11 : 2     Nota 12 : 5  
  
El promedio es : 4.0  
  
Presione una tecla para continuar . . .
```

Observaciones:

- 1) En este ejercicio debemos exhibir las notas y sumarlas al interior del ciclo for porque en cada iteración se almacena una nueva nota en la variable nota, y de ese modo se pierde el valor anterior.
- 2) Dado que la variable nota es entera, debemos hacer "casting" en la línea promedio=(float) suma/n; para que promedio resulte un número real.

22

Escriba una aplicación de consola en C que permita al usuario ejercitar con las tablas de multiplicar del 1 al 12. Los ejercicios se repiten hasta que el usuario termina de ejercitar cuando ingresa como resultado de la multiplicación un - 1.
El programa debe mostrar al final del proceso el porcentaje de aciertos.
Los números a multiplicar deben ser generados aleatoriamente.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    srand(time(NULL));

    int num1, num2, Res, aciertos=0, total=0;

    do
    {
        num1=rand()%12+1; num2=rand()%12+1;
        printf("\n\t%i * %i = ", num1,num2);
        scanf("%i",&Res);
        total++;
        if(Res==num1*num2) aciertos++;
    }
    while(Res!=-1);

    if (total!=1)
        printf("\nEl porcentaje de aciertos es %0.1f%%\n",100.0*aciertos/(total-1));
    else
        printf("\nDebe ingresar al menos un producto\n");

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida

```
8 * 10 = 80
```

```
8 * 9 = 72
```

```
8 * 9 = 72
```

```
2 * 5 = 10
```

```
4 * 1 = 4
```

```
10 * 8 = 80
```

```
12 * 9 = 100
```

```
2 * 9 = -1
```

```
El porcentaje de aciertos es 85.7%
```

```
Presione una tecla para continuar . . .
```

Observaciones:

1) En este ejercicio usamos dos contadores, uno para determinar el total de multiplicaciones realizadas (total) y el otro para contabilizar el total de aciertos (aciertos).

2) Si el total de multiplicaciones es 1, entonces significa que el usuario ingresó al inicio -1, y por lo tanto, terminó con la ejecución del programa; es por esto que se discrimina si el total es igual o distinto de 1.

GUÍA DE PRÁCTICO N° 9

FUNCIONES

Teoría Funciones

36 Funciones

Las *funciones* son un conjunto de instrucciones que realizan una tarea específica, y nos permiten crear módulos reutilizables de código.

Existen funciones predefinidas que están incluidas en las bibliotecas, y funciones definidas por el programador.

Todas las funciones definidas por el programador deben ser declaradas e implementadas.

La declaración de una función se denomina *prototipo*.

La implementación está relacionada con el código que permite realizar la tarea específica de la función.

Sintaxis de la declaración o prototipo:

```
tipo_funcion nombre_funcion(tipo_p1 nombre_p1, ...,tipo_pN nombre_pN);
```

Sintaxis de la implementación o definición:

```
tipo_funcion nombre_funcion(tipo_p1 nombre_p1, ...,tipo_pN nombre_pN)  
{  
  Instrucciones  
}
```

donde

tipo_funcion es el tipo de dato que retorna la función, pudiendo ser void que significa que la función no devuelve un valor

nombre_funcion es el nombre que se le asigna a la función y que debiese ser lo suficientemente descriptivo de la tarea que realiza la función

tipo_px es el tipo de dato asociado al parámetro x-ésimo

nombre_px es el nombre que se le asigna al parámetro x-ésimo

En la declaración de la función es opcional la aparición de los nombres de los parámetros.

La mayoría de las funciones incluyen la orden return que permite devolver o retornar un valor. En la implementación pueden aparecer tantos return como valores devuelva la función.

37 **Función Principal main()**

Todos los programas en C cuentan con, a lo menos, una función y ésta es la función principal main(), que retorna un valor entero igual a 0, si la ejecución ha sido correcta.

Las funciones definidas por el programador pueden ser implementadas antes o después de la función principal main(), pero el prototipo debe aparecer entre la sección de inclusión (include) y la función principal main().

38 **Función Recursiva**

Una función es recursiva cuando se llama a sí misma al interior de su definición. Por ejemplo, la función factorial puede ser considerada recursiva porque $n! = (n - 1)! \cdot n$, es decir, para calcular $n!$ necesitamos calcular $(n - 1)!$

La recursión se ocupa cuando la definición de la función a implementar es recursiva y, por lo tanto, es fácil codificarla.

Ejercicios Funciones

[1] Escriba un programa en C que muestre por pantalla los valores de la función $f(x) = e^{-x} + x^2$, para $x = 1, 1.2, 1.4, \dots, 6$. Defina y use una función apropiada.

[2] a) ¿Qué es una función? ¿Para qué sirve? ¿Cómo se declara una función? ¿Qué es un valor de retorno? ¿Qué son los parámetros o argumentos?
b) Escriba una función que genere un número aleatorio en el intervalo $[a, b]$, donde a y b son los argumentos de la función. Asuma que a y b son números enteros y además $a < b$.

[3] Escriba un programa en C que muestre por pantalla un mensaje indicando si se cumple la relación $|f(x_0) f''(x_0)| < (f'(x_0))^2$ para $f(x) = e^x \text{sen}(2x)$ y para un x_0 ingresado por el usuario. Incorpore en el programa las funciones que considere necesarias.
El programa debe repetir el procedimiento hasta que el usuario ingrese un valor de x_0 que satisfaga la relación dada.

[4] Escriba un programa en C que haciendo uso de una función recursiva muestre por pantalla los primeros 15 números de la sucesión de Fibonacci.
La sucesión de Fibonacci queda definida por :

$$f_0 = 0 ; f_1 = 1 ; f_n = f_{n-1} + f_{n-2}, n = 2, 3, 4, \dots$$

[5] Recordando que $n! = n \cdot (n - 1)!$ escriba una función, definida por usted, que permite calcular y mostrar por pantalla el factorial de un número natural ingresado por el usuario.

[6] Escriba un programa en C que, haciendo uso de una función apropiada, calcule la primera derivada de cuatro funciones polinomiales de la forma $f(x) = ax^5 + bx^4 + cx^3 + dx^2 + ex + f$, donde a, b, c, d, e y f son números reales ingresados por el usuario, al igual que el valor de x . Muestre los resultados usando el circunflejo (^) para representar potencias.

[7] Escriba un programa en C que muestre por pantalla la tabla

angulo	seng(angulo)	cosg(angulo)
0	0.00	1.00
45	0.71	0.71
90	1.00	0.00
180	0.00	-1.00
270	-1.00	0.00

haciendo uso de funciones seng y cosg, que calculan el seno y el coseno del argumento en grados respectivamente, definidas por usted.

[8] Escriba un programa en C que, haciendo uso de una función que determine si un número entero es par o impar, muestre por pantalla la suma de todos los pares y el producto de todos los impares de un conjunto de 150 números aleatorios generados por el computador, entre -67 y 140 inclusive.

[9] Escriba una función que calcule el factorial de un número natural, y use tal función para calcular la suma $\sum_{k=0}^8 \frac{1}{k!}$

[10] Escriba un programa en C que calcule y muestre por pantalla el valor de la función

$$f(x) = \begin{cases} e^{\sqrt{x^2+1}} & , \text{ si } x > \sqrt{7} \\ |2x - 1| & , \text{ si } 0 \leq x \leq \sqrt{7} \\ 10^{-3} \frac{x^4+x^6}{x^2+1} & , \text{ si } x < 0 \end{cases}$$

para un valor de x real ingresado por el usuario. Use una función apropiada definida por usted.

[11] Escriba un programa que determine el menor entero n para el cual la suma $1 + 2 + 3 + \dots + n$ es mayor que 1000. Haga uso de una función apropiada.

[12] Escriba un programa en C que muestre por pantalla, en color, la ecuación y las soluciones de la ecuación cúbica $dx^3 + ax^2 + bx + c = 0$, donde los coeficientes d, a, b y c son ingresados por el usuario, usando el siguiente esquema de trabajo:

Para la ecuación cúbica $x^3 + ax^2 + bx + c = 0$ (si el coeficiente que acompaña a x^3 no es 1, entonces divida la ecuación por tal coeficiente), con $a, b, c \in \mathbb{R}$, primero calcule

$$Q \equiv \frac{a^2 - 3b}{9} \quad R \equiv \frac{2a^3 - 9ab + 27c}{54}$$

Si $R^2 < Q^3$, entonces la ecuación posee tres raíces reales, las cuales son:

$$x_1 = -2\sqrt{Q} \cos\left(\frac{\theta}{3}\right) - \frac{a}{3}$$

$$x_2 = -2\sqrt{Q} \cos\left(\frac{\theta + 2\pi}{3}\right) - \frac{a}{3}$$

$$x_3 = -2\sqrt{Q} \cos\left(\frac{\theta - 2\pi}{3}\right) - \frac{a}{3}$$

donde $\theta = \text{Arccos}\left(\frac{R}{\sqrt{Q^3}}\right)$.

Si $R^2 \geq Q^3$, entonces las tres raíces son:

$$x_1 = (A + B) - \frac{a}{3}$$

$$x_2 = -\frac{1}{2}(A + B) - \frac{a}{3} + i\frac{\sqrt{3}}{2}(A - B)$$

$$x_3 = -\frac{1}{2}(A + B) - \frac{a}{3} - i\frac{\sqrt{3}}{2}(A - B)$$

donde $A = -\text{signo}(R) \left[|R| + \sqrt{R^2 - Q^3} \right]^{1/3}$

$$B = \begin{cases} Q/A, & \text{si } A \neq 0 \\ 0, & \text{si } A = 0 \end{cases}$$

$$\text{signo}(R) = \begin{cases} +1 & , \text{ si } R \geq 0 \\ -1 & , \text{ si } R < 0 \end{cases}$$

Incluya funciones apropiadas.

[13] Escriba un programa en C que calcule y muestre por pantalla el valor de la función

$$f(x) = \begin{cases} \frac{1-x}{\sqrt{x^2-1}} & , \text{ si } x > 1 \\ x+1 & , \text{ si } 0 \leq x \leq 1 \\ \ln(|x|) & , \text{ si } x < 0 \end{cases}$$

para un valor de x real ingresado por el usuario. El programa debe preguntar al usuario si desea repetir la evaluación o si desea terminar. Si el usuario ingresa NO el programa termina, en caso contrario el programa continúa.

Use una función apropiada definida por usted.

[14] Escriba una aplicación de consola que sume y reste dos números enteros aleatorios positivos y menores o iguales a 10.

Defina y use funciones suma y resta apropiadas.

[15] Escriba una aplicación en C que calcule y muestre por pantalla la suma de los primeros 21 polinomios de Laguerre evaluados en x_0 , donde x_0 es ingresado por el usuario.

El programa debe definir y utilizar una función apropiada.

$$(\text{Ind. : } P_0(x) = 1; P_1(x) = 1 - x; P_{n+1}(x) = \frac{(2n+1-x)}{n+1} P_n(x) - \frac{n}{n+1} P_{n-1}(x))$$

[16] Usando una función apropiada, calcule y muestre por pantalla el valor

$$\text{de } f(x, y) = \begin{cases} x^2 + y^2 & , \text{ si } x > y \\ \frac{3}{2}x - 6y & , \text{ en otro caso} \end{cases} \quad \text{donde } x \text{ y } y \text{ son ingresados por el}$$

usuario.

[17] Escriba un programa en C que muestre por pantalla los resultados de evaluar la función

$$f(x, y) = x^2 - 3y^3$$

para $x = m, m + 1, \dots, n - 1, n$ y $y = m, m + 1, \dots, n - 1, n$
donde m y n son números enteros ingresados por el usuario, con $m < n$
Debe implementar una función apropiada.

[18] Use el lenguaje C para mostrar por pantalla los valores de la función $f(x) = \sqrt{x + 3}$ para $x = 0, h, 2h, \dots, xmax$, donde h y $xmax$ son ingresados por el usuario.
Defina y use una función apropiada.

[19] Escriba un programa en C que dibuje N líneas horizontales, la primera de longitud 1, la segunda de longitud 2, hasta la última de longitud N , donde N es un número natural ingresado por el usuario, haciendo uso de una función que dibuje una línea horizontal de tamaño L .

PROBLEMAS RESUELTOS

FUNCIONES

3

Escriba un programa en C que muestre por pantalla un mensaje indicando si se cumple la relación $|f(x_0) f'(x_0)| < (f'(x_0))^2$ para $f(x) = e^x \text{sen}(2x)$ y para un x_0 ingresado por el usuario. Incorpore en el programa las funciones que considere necesarias.

El programa debe repetir el procedimiento hasta que el usuario ingrese un valor de x_0 que satisfaga la relación dada.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

float f(float);
float fp(float);
float fpp(float);

float f(float x)
{
    return exp(x)*sin(2*x);
}

float fp(float x)
{
    return exp(x)*sin(2*x)+2*exp(x)*cos(2*x);
}
```



```

float fpp(float x)
{
    return 4*exp(x)*cos(2*x)-3*exp(x)*sin(2*x);
}

int main()
{
    float x0;

    do
    {
        printf("\nIngrese x0=");
        scanf("%f",&x0);
    }
    while (fabs(f(x0)*fpp(x0))>= fp(x0)*fp(x0));

    printf("\nLa condici%cn se cumple para %0.1f\n",162,x0);

    printf("\n\n");
    system("PAUSE");
    return 0;
}

```

Salida

```

Ingrese x0=6
Ingrese x0=1
Ingrese x0=0.5
La condición se cumple para 0.5
Presione una tecla para continuar . . .

```

Observaciones:

1) En un programa podemos escribir tantas funciones como necesitemos. En este ejercicio se han utilizado tres funciones, una para $f(x)$, otra para $f'(x)$ y la última para $f''(x)$.

2) Las derivadas hemos de calcularlas manualmente.

Tenemos que

$$f(x) = e^x \operatorname{sen}(2x)$$

$$f'(x) = e^x \operatorname{sen}(2x) + 2 e^x \operatorname{cos}(2x)$$

$$f''(x) = e^x \operatorname{sen}(2x) + 2 e^x \operatorname{cos}(2x) + 2 e^x \operatorname{cos}(2x) - 4 e^x \operatorname{sen}(2x) \\ = 4 e^x \operatorname{cos}(2x) - 3 e^x \operatorname{sen}(2x) \quad \square$$

4

Escriba un programa en C que haciendo uso de una función recursiva muestre por pantalla los primeros 15 números de la sucesión de Fibonacci.

La sucesión de Fibonacci queda definida por :

$$f_0 = 0 ; f_1 = 1 ; f_n = f_{n-1} + f_{n-2} , n = 2, 3, 4, \dots$$

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int Fibonacci(int);

int main()
{
    int Total=15;
    int n;
    for (n=0;n<=Total-1;n++)
        printf("Fibonacci(%i)=%i\n",n,Fibonacci(n));
    system("PAUSE");
    return 0;
}

int Fibonacci(int n)
{
    if (n==0) return 0;
```

```
    if (n==1) return 1;
    if (n>=2) return Fibonacci(n-1)+Fibonacci(n-2);
}
```

Salida

```
Fibonacci(0)=0
Fibonacci(1)=1
Fibonacci(2)=1
Fibonacci(3)=2
Fibonacci(4)=3
Fibonacci(5)=5
Fibonacci(6)=8
Fibonacci(7)=13
Fibonacci(8)=21
Fibonacci(9)=34
Fibonacci(10)=55
Fibonacci(11)=89
Fibonacci(12)=144
Fibonacci(13)=233
Fibonacci(14)=377
Presione una tecla para continuar . . .
```

Observaciones:

- 1) La implementación de una función puede aparecer antes o después de la función principal main().
- 2) La función de Fibonacci es un ejemplo de función recursiva, porque en su definición se hacen llamados a la misma función.
- 3) En una función se pueden incluir tantos return como valores puede entregar la función.

8

Escriba un programa en C que, haciendo uso de una función que determine si un número entero es par o impar, muestre por pantalla la suma de todos los pares y el producto de todos los impares de un conjunto de 25 números aleatorios generados por el computador, entre - 67 y 140 inclusive.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int par(int);

int par(int x)
{
    if(x%2==0) return 1;
    else return 0;
}

int main()
{
    int i, num, sumaPar=0, prodImpar=1;
    srand(time(NULL));
    for(i=1;i<=25;i++)
    {
        num=rand()%208-67;
        printf("\nnum %i = %i\n",i,num);
        if (par(num)==1) sumaPar=sumaPar+num;
        else if(num%3==0) prodImpar=prodImpar*num;
    }
    printf("\nLa suma de los pares es %i\n",sumaPar);
    printf("\nEl producto de los impares es %i\n",prodImpar);

    system("PAUSE");
    return 0;
} □
```

Salida

```
num 1 = 76
num 2 = 11
num 3 = 88
num 4 = -57
num 5 = 23
num 6 = 62
num 7 = -10
num 8 = 51
num 9 = -34
num 10 = -8
num 11 = 90
num 12 = 116
num 13 = 7
num 14 = -54
num 15 = -66
num 16 = 42
num 17 = 19
num 18 = 127
num 19 = -28
num 20 = 61
num 21 = 54
num 22 = 59
num 23 = 26
num 24 = -42
num 25 = 57
La suma de los pares es 312
El producto de los impares es -165699
Presione una tecla para continuar . . .
```

Observación:

En este ejercicio se asumió que si el número es par, entonces la función devuelve un 1, y si es impar devuelve un 0. Esta asignación de valores es

arbitraria, esto significa que podemos relacionar par e impar con los valores que queramos. \square

12

Escriba un programa en C que muestre por pantalla, en color, la ecuación y las soluciones de la ecuación cúbica $dx^3 + ax^2 + bx + c = 0$, donde los coeficientes d, a, b y c son ingresados por el usuario, usando el siguiente esquema de trabajo:

Para la ecuación cúbica $x^3 + ax^2 + bx + c = 0$ (si el coeficiente que acompaña a x^3 no es 1, entonces divida la ecuación por tal coeficiente), con $a, b, c \in \mathbb{R}$, primero calcule

$$Q \equiv \frac{a^2 - 3b}{9} \quad R \equiv \frac{2a^3 - 9ab + 27c}{54}$$

Si $R^2 < Q^3$, entonces la ecuación posee tres raíces reales, las cuales son:

$$x_1 = -2\sqrt{Q} \cos\left(\frac{\theta}{3}\right) - \frac{a}{3}$$

$$x_2 = -2\sqrt{Q} \cos\left(\frac{\theta + 2\pi}{3}\right) - \frac{a}{3}$$

$$x_3 = -2\sqrt{Q} \cos\left(\frac{\theta - 2\pi}{3}\right) - \frac{a}{3}$$

donde $\theta = \text{Arccos}\left(\frac{R}{\sqrt{Q^3}}\right)$.

Si $R^2 \geq Q^3$, entonces las tres raíces son:

$$x_1 = (A + B) - \frac{a}{3}$$

$$x_2 = -\frac{1}{2}(A + B) - \frac{a}{3} + i\frac{\sqrt{3}}{2}(A - B)$$

$$x_3 = -\frac{1}{2}(A + B) - \frac{a}{3} - i\frac{\sqrt{3}}{2}(A - B)$$

donde $A = -\text{signo}(R) \left[|R| + \sqrt{R^2 - Q^3} \right]^{1/3}$

$$B = \begin{cases} Q/A, & \text{si } A \neq 0 \\ 0, & \text{si } A = 0 \end{cases}$$

$$\text{signo}(R) = \begin{cases} +1 & , \text{ si } R \geq 0 \\ -1 & , \text{ si } R < 0 \end{cases}$$

Incluya funciones apropiadas.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <conio2.h>

int signo(float);

int main()
{
    float a, b, c, d;
    float Q, R;
    float angulo;
    float x1, x2, x3;
    float x2r, x2i, x3r, x3i;
    float A, B;

    printf("\nIngrese d = ");
    scanf("%f",&d);
    printf("\nIngrese a = ");
    scanf("%f",&a);
    printf("\nIngrese b = ");
    scanf("%f",&b);
    printf("\nIngrese c = ");
    scanf("%f",&c);
```

```

textbackground(WHITE);

if (d!=0.0)
{
  if (d!=1)
  {
    a=a/d; b=b/d; c=c/d;
  }
  Q=(a*a-3*b)/9.0; R=(2*pow(a,3)-9*a*b+27*c)/54;
  angulo=acos(R/pow(sqrt(Q),3));
  if (R*R<pow(Q,3))
  {
    x1=-2*sqrt(Q)*cos(angulo/3.0)-a/3;
    x2=-2*sqrt(Q)*cos((angulo+2*M_PI)/3.0)-a/3;
    x3=-2*sqrt(Q)*cos((angulo-2*M_PI)/3.0)-a/3;
    textcolor(RED);
    printf("\nx1 = %0.5f\nx2 = %0.5f\nx3 = %0.5f\n",x1,x2,x3);
  }
  else
  {
    A=-signo(R)*pow((fabs(R)+sqrt(R*R-pow(Q,3))),1/3.);
    if (A!=0.0)
      B=Q/A;
    else
      B=0;
    x1=(A+B)-a/3.;
    x2r=-1/2.*(A+B)-a/3.;
    x2i=sqrt(3)/2*(A-B);
    x3r=x2r;
    x3i=-x2i;
    textcolor(CYAN);
    printf("\nx1 = %0.5f\n",x1);
    printf("\nx2 = %0.5f + %0.5f i\n",x2r,fabs(x2i));
    printf("\nx3 = %0.5f - %0.5f i\n",x3r,fabs(x3i));
  }
}
printf("\n\n");
system("PAUSE");
return 0;
}

```



```
int signo(float x)
{
    if (x>=0.0) return 1;
    else return -1;
}
```

Salida

```
Ingrese d = -3
Ingrese a = 2
Ingrese b = 1
Ingrese c = -5
x1 = -1.07768
x2 = 0.87218 + 0.88647 i
x3 = 0.87218 - 0.88647 i
Presione una tecla para continuar . . .
```

Observaciones:

1) En este ejercicio se asocia, arbitrariamente, signo positivo con el 1 y signo negativo con el -1 .

2) Recordemos que si una función polinomial posee raíces complejas, estas aparecen de a pares, es decir, si $m + ni$ es raíz, entonces $m - ni$ también lo será.

Escriba un programa en C que calcule y muestre por pantalla el valor de la función

$$f(x) = \begin{cases} \frac{1-x}{\sqrt{x^2-1}} & , \text{ si } x > 1 \\ x + 1 & , \text{ si } 0 \leq x \leq 1 \\ \ln(|x|) & , \text{ si } x < 0 \end{cases}$$

para un valor de x real ingresado por el usuario. El programa debe preguntar al usuario si desea repetir la evaluación o si desea terminar. Si el usuario ingresa NO el programa termina, en caso contrario el programa continúa. Use una función apropiada definida por usted.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>

float f(double);

float f(double x)
{
    if (x>1.0) return ((1-x)/sqrt(x*x-1));
    else if ((x>=0.0) && (x<=1.0)) return (x+1);
    else return (log(fabs(x)));
}

int main()
{
    float valor;
    char Resp[3];
    do
    {
        printf("\nIngrese un valor real en donde evaluar la funci%cn: ",162);
        scanf("%f",&valor);
```

```

printf("\nf(%.2f) = %.2f\n",valor,f(valor));
printf("\nDesea continuar (SI o NO) : ");
scanf("%s",Resp);
}
while(strcmp(Resp,"NO")!=0);

printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida

```

Ingrese un valor real en donde evaluar la función: 0.5
f<0.50> = 1.50
Desea continuar <SI o NO> : SI
Ingrese un valor real en donde evaluar la función: 1.2
f<1.20> = -0.30
Desea continuar <SI o NO> : SI
Ingrese un valor real en donde evaluar la función: -2.5
f<-2.50> = 0.92
Desea continuar <SI o NO> : NO

Presione una tecla para continuar . . .

```

Observaciones:

- 1) En este ejercicio se asume que el usuario ingresa NO sólo en mayúsculas.
- 2) El bucle do...while permite repetir el proceso hasta que el usuario ingrese NO.
- 3) El tipo de la variable x es double pudiendo haber sido float, la diferencia es sólo el intervalo en que se mueve cada tipo (Ver 19)

Escriba una aplicación de consola que sume y reste dos números enteros aleatorios positivos y menores o iguales a 10.
Defina y use funciones suma y resta apropiadas.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int suma(int, int);

int suma(int a,int b)
{
    return a+b;
}

int resta(int,int);

int resta(int a, int b)
{
    return a-b;
}

int main()
{

    srand(time(NULL));
    int x,y;
    x=rand()%10+1;
    y=rand()%10+1;
    printf("\n%i + %i = %i\n",x,y,suma(x,y));
    printf("\n%i - %i = %i\n",x,y,resta(x,y));

    printf("\n\n");
```

```
system("PAUSE");  
return 0;  
}
```

Salida

```
4 + 1 = 5  
4 - 1 = 3  
  
Presione una tecla para continuar . . .
```

Observación:

La función suma posee dos argumentos de tipo entero.

15

Escriba una aplicación en C que calcule y muestre por pantalla la suma de los primeros 21 polinomios de Laguerre evaluados en x_0 , donde x_0 es ingresado por el usuario.

El programa debe definir y utilizar una función apropiada.

(Ind. : $P_0(x) = 1$; $P_1(x) = 1 - x$; $P_{n+1}(x) = \frac{(2n+1-x)}{n+1} P_n(x) - \frac{n}{n+1} P_{n-1}(x)$)

Solución:

Código Fuente

```
#include <stdio.h>  
#include <stdlib.h>
```

```
float P(float, int);
```

```

float P(float x, int n)
{
    if(n==0) return 1;
    else if(n==1) return 1-x;
    else return (2*n-1.-x)/n*P(x,n-1)-(n-1.)/n*P(x,n-2);
}

int main()
{
    float x0, suma=0.0;
    int n;

    printf("\nIngrese x0 : ");
    scanf("%f",&x0);

    for(n=0;n<=20;n++)
        suma=suma+P(x0,n);

    printf("\nLa suma de los 21 primeros polinomios evaluados");
    printf(" en x0 = %0.1f es %0.4f\n",x0,suma);

    printf("\n\n");
    system("PAUSE");
    return 0;
}

```

Salida

```

Ingrese x0 : 0.8
La suma de los 21 primeros polinomios evaluados en x0 = 0.8 es 1.9641
Presione una tecla para continuar . . .

```

Observación:

Los polinomios de Laguerre se obtienen mediante el uso de una función recursiva en el programa, porque es fácil implementar la función dadas las relaciones

$$P_0(x) = 1; P_1(x) = 1 - x; P_{n+1}(x) = \frac{(2n+1-x)}{n+1} P_n(x) - \frac{n}{n+1} P_{n-1}(x) \quad \square$$

16

Usando una función apropiada, calcule y muestre por pantalla el valor de

$$f(x, y) = \begin{cases} x^2 + y^2 & , \text{ si } x > y \\ \frac{3}{2}x - 6y & , \text{ en otro caso} \end{cases} \quad \text{donde } x \text{ y } y \text{ son ingresados por el usuario.}$$

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

float f(float , float);

float f(float x, float y)
{
    if(x>y) return x*x+y*y;
    else return 3./2*x-6*y;
}

int main()
{
    float x,y;

    printf("\nIngrese x : ");
    scanf("%f",&x);
```

```

printf("\nIngrese y : ");
scanf("%f",&y);

printf("\n f(%0.1f,%0.1f) = %0.4f\n",x,y,f(x,y));

printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida 1

```

Ingrese x : 0.6
Ingrese y : 3
f<0.6,3.0> = -17.1000
Presione una tecla para continuar . . .

```

Salida 2

```

Ingrese x : 4
Ingrese y : 1
f<4.0,1.0> = 17.0000
Presione una tecla para continuar . . .

```



17

Escriba un programa en C que muestre por pantalla los resultados de evaluar la función

$$f(x, y) = x^2 - 3y^3$$

para $x = m, m + 1, \dots, n - 1, n$ y $y = m, m + 1, \dots, n - 1, n$
donde m y n son números enteros ingresados por el usuario, con $m < n$
Debe implementar una función apropiada.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int f(int x, int y);

int f(int x, int y)
{
    return pow(x,2)-3*pow(y,3);
}

int main()
{
    int x,y;
    int m,n;
    do
    {
        printf("\nIngrese m : ");
        scanf("%i",&m);
        printf("\nIngrese n : ");
        scanf("%i",&n);
    }
    while(m>=n);
    for(x=m;x<=n;x++)
        for(y=m;y<=n;y++)
            printf("\nf(%i,%i) = %i\n",x,y,f(x,y));

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida

```
Ingrese m : 3
Ingrese n : 1
Ingrese m : 1
Ingrese n : 3
f(1,1) = -2
f(1,2) = -23
f(1,3) = -80
f(2,1) = 1
f(2,2) = -20
f(2,3) = -77
f(3,1) = 6
f(3,2) = -15
f(3,3) = -72

Presione una tecla para continuar . . .
```

Observaciones:

- 1) Si m y n son enteros, entonces los valores de x y y también son enteros de acuerdo al enunciado.
- 2) En el prototipo o declaración de la función se indican los nombres de los argumentos, pero recordemos que esto es opcional.

18

Use el lenguaje C para mostrar por pantalla los valores de la función $f(x) = \sqrt{x+3}$ para $x = 0, h, 2h, \dots, xmax$, donde h y $xmax$ son ingresados por el usuario.

Defina y use una función apropiada.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

float f(float); // declaración de f

int main()
{
    float xmax, h;
    float x;
    printf("Ingrese el l%cmite superior del intervalo : ",161);
    scanf("%f",&xmax);
    printf("Ingrese el tama%co de paso h : ",164);
    scanf("%f",&h);
    printf("\n");
    for (x=0;x<=xmax;x=x+h)
        printf("f(%0.2f)= %0.2f\n",x,f(x));

    printf("\n\n");
    system("PAUSE");
    return 0;
}

float f(float x) // definición de f
{
    return sqrt(x+3);
}
```

Salida 1

```
Ingrese el límite superior del intervalo : 5
Ingrese el tamaño de paso h : 0.5
f(0.00)= 1.73
f(0.50)= 1.87
f(1.00)= 2.00
f(1.50)= 2.12
f(2.00)= 2.24
f(2.50)= 2.35
f(3.00)= 2.45
f(3.50)= 2.55
f(4.00)= 2.65
f(4.50)= 2.74
f(5.00)= 2.83

Presione una tecla para continuar . . .
```

Salida 2

```
Ingrese el límite superior del intervalo : 1
Ingrese el tamaño de paso h : 1
f(0.00)= 1.73
f(1.00)= 2.00

Presione una tecla para continuar . . .
```

Observaciones:

- 1) En este programa se asume que el usuario ingresa valores apropiados para h y x_{max} .
- 2) Si el usuario ingresa un valor de x_{max} menor que h , entonces la salida es

```
Ingrese el límite superior del intervalo : 1
Ingrese el tamaño de paso h : 4
f(0.00)= 1.73

Presione una tecla para continuar . . .
```

Es decir, se evalúa la función sólo en $x = 0$.



Escriba un programa en C que dibuje N líneas horizontales, la primera de longitud 1, la segunda de longitud 2, hasta la última de longitud N, donde N es un número natural ingresado por el usuario, haciendo uso de una función que dibuje una línea horizontal de tamaño L.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

void LineaHorizontal(int);

void LineaHorizontal(int L)
{
    int i;
    for (i=1;i<=L;i++)
        printf("%c",196);
    printf("\n");
}

int main()
{
    int i, N;
    printf("Ingrese el número de líneas horizontales N : ");
    scanf("%i",&N);
    for (i=1;i<=N;i++)
        LineaHorizontal(i);

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```


GUÍA DE PRÁCTICO N° 10

TIPOS ESTRUCTURADOS

Teoría Tipos Estructurados

T 39 Tipos Estructurados

Los *tipos de datos estructurados* o *tipos compuestos* son agrupaciones de otros tipos de datos, habitualmente básicos.

Ejemplos de tipos estructurados en C son los arreglos unidimensionales o vectores y los arreglos bidimensionales o matrices.

Para manejar un arreglo, las etapas a considerar son:

- 1) Declaración del arreglo
- 2) Creación del arreglo
- 3) Inicialización de los elementos del arreglo
- 4) Acceso a los elementos del arreglo.

T 40 Arreglos Unidimensionales (Vectores)

Los arreglos (arrays) unidimensionales o vectores permiten agrupar elementos del mismo tipo, que pueden ser referenciados con el mismo nombre, y hacen uso de un índice para indicar la posición de una componente.

Por ejemplo, podemos declarar un vector llamado Altura que almacene las alturas de 7 personas de la siguiente manera

```
float Altura[7];
```

Para acceder a la altura de la primera persona registrada debemos escribir `Altura[0]`, para la altura de la segunda persona `Altura[1]`, y así sucesivamente hasta llegar a la altura de la séptima persona que será `Altura[6]`.

Si es necesario, podríamos inicializar el vector `Altura`, por ejemplo, de la siguiente manera:

```
float Altura[7] = {35.7,47.5,98.4,75,108.2,45.3,77.7};
```

Los arreglos en C siempre deben comenzar con el índice 0, y este índice, que indica posición, debe ser incrementado en una unidad cada vez, hasta llegar al total de componentes menos 1. Observamos además, que en el ejemplo todas las alturas son del mismo tipo, es decir, números reales (`float`).

Los arreglos unidimensionales requieren del uso de, al menos, una estructura iterativa para acceder a las componentes de este; la estructura iterativa que se ocupa con frecuencia es el `for`.

41 **Arreglos Bidimensionales (Matrices)**

Los arreglos (arrays) bidimensionales o matrices permiten trabajar elementos del mismo tipo, que pueden ser referenciados con el mismo nombre, y a diferencia de los vectores, que hacen uso de un solo índice, las matrices utilizan dos índices, uno para la fila y otro para la columna.

Por ejemplo, podemos declarar una matriz llamada `Inversa` que almacene la matriz inversa de una matriz dada de orden 2×2 , cuyos valores son reales, de la siguiente manera

```
float Inversa[2][2];
```

El primer índice, se relaciona con las filas de la inversa y el segundo índice con las columnas. Así, por ejemplo, `Inversa[0][0]` representa el elemento que está en la primera fila y en la primera columna de la inversa.

Si es necesario, podríamos inicializar la matriz `Inversa`, por ejemplo, de la siguiente manera:

```
float Inversa[2][2] = {{2.1,7},{5,9.6}};
```


Los arreglos bidimensionales requieren del uso de, al menos, dos estructuras iterativas para acceder a las componentes del arreglo. La estructura iterativa que se ocupa con frecuencia es el for; un for para el primer índice y otro for para el segundo índice.

42 **Arreglos Multidimensionales**

El lenguaje C, al igual que muchos otros lenguajes, permiten el uso de arreglos multidimensionales, es decir, con más de dos índices; pero éstos ocupan mucha memoria por lo que no son ampliamente utilizados.

Por ejemplo, podemos declarar un arreglo de la siguiente manera

```
float Tridimensional[20][20][20];
```

que podría representar las coordenadas de algún objeto en el espacio.

Ejercicios

Tipos Estructurados

- [1] Escriba un programa en C que solicite al usuario el ingreso de las notas de la primera prueba de 5 alumnos de un curso y muestre por pantalla, con 1 decimal, el promedio y la desviación estándar de las notas del curso.
- [2] Un curso tiene 100 alumnos y cada alumno tiene 3 notas con igual ponderación cada una de ellas. Use C para calcular y mostrar por pantalla el promedio para cada alumno, sabiendo que las notas del 1 al 7 con un decimal, son generadas aleatoriamente.
- [3] Escriba un programa en C que calcule el promedio y la desviación estándar de n notas del 1 al 7, donde n es un número natural ingresado por el usuario, al igual que las notas.
- [4] Use el lenguaje C para calcular la norma de un vector de \mathbb{R}^n , donde n es un número natural menor o igual que 3 ingresado por el usuario. Las componentes del vector son generadas aleatoriamente.
- [5] Use C para determinar si un vector de \mathbb{R}^n es unitario o no. Las componentes del vector son ingresadas por el usuario, con n un número natural aleatorio no mayor a 20.
- [6] Escriba un programa en C que muestre por pantalla el producto interior de dos vectores de 15 componentes enteras ingresadas por el usuario, e indique si los vectores son ortogonales o no.

[7] Usando vectores, calcule y muestre por pantalla, en grados, el ángulo entre los vectores $\mathbf{a} = [a_1, a_2, a_3]$ y $\mathbf{b} = [b_1, b_2, b_3]$, donde \mathbf{a} y \mathbf{b} son ingresados por el usuario.

[8] Escriba un programa en C que calcule la norma euclidiana del vector
$$\begin{bmatrix} 1 \\ -2 \\ 5 \end{bmatrix}$$

[9] Escriba un programa en C que calcule la distancia de la recta
$$L : \begin{cases} x = x_0 + at \\ y = y_0 + bt \\ z = z_0 + ct \end{cases}$$

al punto $P(1, -2, 0)$, donde (x_0, y_0, z_0) es ingresado por el usuario al igual que el vector director $[a, b, c] \neq [0, 0, 0]$

[10] Escriba un programa en C que solicite al usuario el ingreso las componentes de un vector $(v_1, v_2) \in \mathbb{R}^2$ y que muestre por pantalla, con 2 decimales, la norma del máximo. Además, el programa debe mostrar un mensaje en pantalla indicando cuando la norma es mayor o igual que 1. (Obs.: La norma del máximo es : $\|(v_1, v_2)\| = \max\{|v_1|, |v_2|\}$)

[11] Escriba un programa en C que solicite al usuario el ingreso de dos vectores \mathbf{a} y \mathbf{b} , y luego muestre por pantalla el producto vectorial $\mathbf{a} \times \mathbf{b}$. El programa debe mostrar además un mensaje indicando si los vectores son paralelos o no.

[12] Escriba un programa en C que muestre por pantalla el producto escalar triple $\langle \mathbf{a}, \mathbf{b} \times \mathbf{c} \rangle$, donde las componentes de los vectores son ingresadas por el usuario.

[13] Sean \mathbf{a} y \mathbf{b} dos vectores no nulos de \mathbb{R}^n . Se llama proyección de \mathbf{b} sobre \mathbf{a} al vector:

$$\text{proy}_{\mathbf{a}} \mathbf{b} = \frac{\langle \mathbf{a}, \mathbf{b} \rangle}{\|\mathbf{a}\|^2} \mathbf{a}$$

Escriba un programa en C que muestre por pantalla la proyección de \mathbf{b} sobre \mathbf{a} , donde \mathbf{a} y \mathbf{b} son ingresados por el usuario al igual que el valor de n .

[14] Use el lenguaje de programación C para verificar si el conjunto $B = \{[a_1, a_2, a_3], [b_1, b_2, b_3], [c_1, c_2, c_3]\}$ es ortonormal o no, donde las componentes de los vectores del conjunto son ingresadas por el usuario.

[15] Escriba un programa en C que lea por pantalla el valor de n y las n componentes reales de un vector \mathbf{x} . Luego muestre por pantalla lo siguiente:

- i) el vector \mathbf{x}
- ii) la norma euclidiana de \mathbf{x}
- iii) la suma de los elementos de índice impar
- iv) un mensaje indicando si el promedio de los elementos de índice par es mayor o igual que 5.

Separe con una línea horizontal cada ítem anterior.

Obs.: Suponga que la primera componente de \mathbf{x} es $\mathbf{x}[1]$

[16] Use C para determinar el mayor de n números ingresados por el usuario, donde n es un entero aleatorio en el intervalo $[1, 25]$.

[17] Escriba un programa en C que genere en forma aleatoria m notas reales, del 1 al 7 con un decimal, para cada uno de los n alumnos de un curso, mostrando por pantalla los alumnos aprobados (promedio ≥ 4) junto con su promedio, usando dos decimales. Numere los alumnos del 1 al n para identificarlos.

Los valores m y n son ingresados por el usuario, quien debe ser obligado a ingresar números naturales mayores o iguales que 4, en ambos casos.

El programa además debe mostrar por pantalla el número de alumnos con distinción (promedio $\in [5, 5.6)$) y distinción máxima (promedio ≥ 5.7).

[18] El coeficiente de correlación nos entrega una medida de cómo dos variables aleatorias están asociadas.

Si $\vec{x} = (x_1, x_2, x_3, \dots, x_n)$ y $\vec{y} = (y_1, y_2, y_3, \dots, y_n)$ son vectores que almacenan los datos asociados a las variables, entonces el coeficiente de correlación está dado por la fórmula:

$$r = \frac{\sum_{i=1}^n x_i y_i - n \bar{x} \bar{y}}{\sqrt{\left(\sum_{i=1}^n x_i^2 - n (\bar{x})^2 \right) \cdot \left(\sum_{i=1}^n y_i^2 - n (\bar{y})^2 \right)}}$$

$$\text{con } \bar{x} = \frac{\sum_{i=1}^n x_i}{n} \text{ y } \bar{y} = \frac{\sum_{i=1}^n y_i}{n}$$

Escriba un programa en C que calcule y muestre por pantalla el coeficiente de correlación de dos vectores de n componentes reales generadas aleatoriamente, donde n es ingresado por el usuario. Suponga que las componentes de los vectores son números reales con un decimal, en el intervalo $[-100, 100]$.

[19] Use C para obtener, a partir del ingreso por parte del usuario de a lo más 20 puntos, la ecuación de la recta de regresión $y = a + bx$, es decir los valores de a y b , usando las siguientes fórmulas:

$$a = \bar{y} - b \bar{x} ; b = \frac{S_{XY}}{S_X^2}$$

$$\text{donde } \bar{y} = \frac{\sum_{i=1}^n y_i}{n} ; \bar{x} = \frac{\sum_{i=1}^n x_i}{n} ; S_{XY} = \frac{\sum_{i=1}^n x_i y_i}{n} - \bar{x} \bar{y} ; S_X^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$$

Además el programa debe obtener el coeficiente de correlación $r = \frac{S_{XY}}{S_X \cdot S_Y}$ y el coeficiente de determinación, que sirve para determinar la bondad del ajuste de y en función de x , que es $R^2 = r^2$.

$$\text{Observe que: } S_Y = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2}$$

[20] Suponga que se han obtenido experimentalmente n parejas de valores (x_i, y_i) , y se pretende encontrar la ecuación de una recta $y = a + bx$ que representa la dependencia lineal de y con respecto a x ; esta recta se

denomina recta de regresión lineal de y sobre x . Para obtener los valores de a y b se aplican las siguientes fórmulas:

$$a = \bar{y} - b \bar{x} ; b = \frac{S_{XY}}{S_X^2}$$

$$\text{donde } \bar{y} = \frac{\sum_{i=1}^n y_i}{n} ; \bar{x} = \frac{\sum_{i=1}^n x_i}{n} ; S_{XY} = \frac{\sum_{i=1}^n x_i y_i}{n} - \bar{x} \bar{y} ; S_X^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$$

Escriba un programa en C que lea el valor de n y los n pares (x_i, y_i) , mostrando por pantalla la ecuación de la recta de regresión obtenida, permitiendo al usuario volver a ingresar el valor de n y los pares de puntos hasta tres veces seguidas.

[21] Escriba un programa en C que permita el ingreso de los coeficientes del polinomio $ax^4 + bx^3 + cx^2 + dx + e$, el ingreso de un valor inicial x_0 y de un error (que es un número positivo pequeño) y calcule una raíz usando el método de Newton-Raphson que consiste en obtener aproximaciones sucesivas x_{n+1} usando la fórmula:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, n = 0, 1, 2, \dots$$

donde x_0 , como ya se dijo, es un valor inicial ingresado por el usuario, pero que debe cumplir la condición

$$|f(x_0) \cdot f''(x_0)| < (f'(x_0))^2$$

Recuerde que $f'(x)$ y $f''(x)$ representa la primera y la segunda derivada de $f(x)$, respectivamente.

[22] El método de Runge-Kutta para resolver la ecuación diferencial

$$\frac{dy}{dt} = f(t, y), a \leq t \leq b$$

$$y(a) = \alpha$$

consiste en obtener valores $y_i \equiv y(t_i)$ usando

$$y_0 = \alpha$$

$$k_1 = f(t_i, y_i)$$

$$k_2 = f(t_i + \frac{h}{2}, y_i + \frac{h}{2} k_1)$$

$$k_3 = f\left(t_i + \frac{h}{2}, y_i + \frac{h}{2}k_2\right)$$

$$k_4 = f(t_i + h, y_i + hk_3)$$

$$y_{i+1} = y_i + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4), \quad i = 0, 1, \dots, N - 1$$

donde

$$h = \frac{b-a}{N} ; t_i = a + ih \text{ con } i = 0, 1, 2, \dots, N$$

Resuelva, usando C, la ecuación diferencial

$$\frac{dy}{dt} = y - t^2 + 1, \quad a \leq t \leq b$$

$$y(a) = 0.5$$

con $N = 20$ por medio del método de Runge-Kutta, donde a y b son ingresados por el usuario, con a menor que b . Considere 6 decimales.

[23] Prepare una aplicación en C que genere una tabla con tiempo t_i , población observada y_i , población calculada \hat{y}_i y % de error, donde

$$\% \text{ error} = \frac{\text{pob. observada} - \text{pob. calculada}}{\text{pob. observada}} \times 100$$

Se dispone de los siguientes datos observados:

Años (t_i)	Número de Lechuzas (y_i)
0	5000
2	60000
4	10000
6	5000
8	70000
10	5000
12	10000

La población calculada (\hat{y}_i) se obtiene haciendo uso de un modelo exponencial de la forma $\hat{y}_i = c e^{bt_i}$, donde

$$c = e^a$$

$$a = \frac{1}{n} \sum_{i=1}^n \ln(y_i) - \frac{1}{n} \sum_{i=1}^n t_i$$

$$b = \frac{\sum_{i=1}^n t_i \ln(y_i) - \frac{\sum_{i=1}^n t_i \sum_{i=1}^n \ln(y_i)}{n}}{\sum_{i=1}^n t_i^2 - \frac{\left(\sum_{i=1}^n t_i\right)^2}{n}}$$

y n representa el número de datos.

[24] Prepare un programa en C que genere una tabla de doble entrada $x \setminus n$ para los polinomios de Legendre $P_n(x)$ definidos por la relación de recurrencia

$$P_{n+1}(x) = \frac{2n+1}{n+1} x P_n(x) - \frac{n}{n+1} P_{n-1}(x)$$

$$P_0(x) = 1$$

$$P_1(x) = x$$

con $n = 0, 1, 2, 3$ y $x \in [a, b]$ donde a y b son ingresados por el usuario. Considere 30 subdivisiones del intervalo anterior.

[25] a) Escriba un programa en C que solicite al usuario el ingreso de los elementos de una matriz A de orden 2×3 y muestre por pantalla la matriz

$$A \cdot \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$$

b) Muestre por pantalla, usando formato matricial, la matriz de orden 6

$$A = (a_{ij}) = \begin{cases} \frac{1}{i+j} & , \text{ si } i - j < 0 \\ i^j & , \text{ si } i \geq j \end{cases}$$

[26] Escriba un programa en C que calcule y muestre por pantalla, en formato matricial, la suma $3A + 2B$, donde A y B son matrices cuadradas de orden 6 ingresadas por el usuario.

El programa además debe mostrar por pantalla la traza de $3A + 2B$.

[27] Escriba un programa en C que muestre por pantalla, usando

formato matricial, el producto $A \cdot B$ donde $A = \begin{bmatrix} 1 & -1 & 2 \\ 2 & 0 & 1 \\ 1 & -5 & 3 \end{bmatrix}$ y $B = (b_{ij})_3$,

con $b_{ij} = \frac{1}{i+j}$.

[28] Muestre por pantalla la norma euclidiana de la matriz de orden 2×3 cuyos elementos son todos números no negativos menores o iguales que 1 y generados al azar.

[29] Escriba un programa en C que muestre por pantalla el determinante de una matriz de orden 3 cuyos elementos son números aleatorios enteros entre -1 y 1 , incluyendo los extremos.

[30] Escriba un programa que calcule y muestre por pantalla el determinante de una matriz de orden 3 cuyos elementos son ingresados por el usuario.

[31] Use C para calcular y mostrar por pantalla los valores propios de una matriz $A \in \mathcal{M}_2(\mathbb{R})$ cuyos elementos son ingresados por el usuario.

[32] Escriba un programa en C que muestre por pantalla, en formato matricial, la traspuesta de una matriz cuadrada de orden 3.

[33] Use C para mostrar por pantalla si una matriz de orden 3, ingresada por el usuario, es antisimétrica o no.

[34] Escriba un programa en C que muestre por pantalla, usando formato matricial, el producto $A^T \cdot A$, donde $A = (a_{ij})$ con $a_{ij} = (-1)^{i+j}$, $i = 1, 2, 3$; $j = 1, 2, 3$.

[35] Escriba un programa en C que muestre por pantalla, usando formato matricial, el producto $A^T \cdot A$, donde $A = (a_{ij}) \in \mathcal{M}_n(\mathbb{R})$ con $a_{ij} = (-1)^{i+j}$ y n ingresado por el usuario.

[36] Use C para calcular y mostrar por pantalla la norma euclidiana de la matriz $A = (a_{ij})_3$, donde $a_{ij} = \begin{cases} (-1)^{i+j} & , \text{ si } i > j \\ 2^j & , \text{ si } i \leq j \end{cases}$

[37] Escriba un programa en C que muestre por pantalla, con 2 decimales, los elementos de $A + B$ donde $A = (a_{ij})$ y $B = I$ con $a_{ij} = i - j$; $A, B \in \mathcal{M}_2(\mathbb{R})$

[38] Escriba un programa en C que calcule y muestre por pantalla, en formato matricial, el producto de dos matrices de orden 2, cuyos elementos son ingresados por el usuario.

[39] Escriba un programa en C que genere aleatoriamente los elementos de una matriz cuadrada A de orden 3 y muestre por pantalla el resultado de $A^T A$ y $A A^T$. Los elementos de la matriz A deben ser números enteros, en el rango $[-5, 5]$

[40] Escriba un programa en C que calcule el producto $A \cdot B$, con $A \in \mathcal{M}_{m \times p}(\mathbb{R})$ y $B \in \mathcal{M}_{p \times n}(\mathbb{R})$. El usuario es quien ingresa el orden y los elementos de cada matriz.

[41] Prepare un programa en C que solicite al usuario el ingreso de una matriz triangular inferior de orden 3, y calcule la traza y la inversa de la matriz ingresada, exhibiendo estos resultados por pantalla con 1 decimal usando un color diferente para cada resultado.

Los ceros de la matriz triangular inferior, es decir los que están por sobre la diagonal principal, deben ser asignados en forma automática por el programa, no solicitándole al usuario el ingreso de éstos.

[42] Escriba un programa en C que muestre por pantalla, en formato matricial, el producto de dos matrices antisimétricas de orden n ingresadas por el usuario, donde n es un entero positivo del 1 al 10 generado al azar.

[43] Escriba un programa en C que calcule la traza de una matriz de orden n , cuyos elementos enteros son generados aleatoriamente en el rango $[-10, 15]$. El orden de la matriz es ingresado por el usuario.

[44] Los polinomios de Legendre se definen recursivamente por $P_0(x) = 1$, $P_1(x) = x$ y $P_{n+2}(x) = \frac{2n+3}{n+2}x P_{n+1}(x) - \frac{n+1}{n+2}P_n(x)$, $n \geq 0$.

Sea $M = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix}$, donde $m_{ij} = P_{i+j}(1)$.

Escriba un programa en C que genere la matriz M y, calcule y muestre por pantalla su inversa. Use funciones apropiadas.

[45] Escriba una aplicación en C, que calcule y muestre por pantalla la traza de una matriz cuadrada, cuyo orden es un número ingresado por el

usuario, y cuyos elementos son números aleatorios impares en el intervalo $[-3, 13]$.

El programa debe obligar al usuario a ingresar un orden mayor o igual que 2.

[46] Escriba un programa en C que solicite al usuario el ingreso del orden y de los elementos de una matriz $A \in \mathcal{M}_n(\mathbb{R})$, y que muestre por pantalla :

a) La suma de los elementos extradiagonales.

b) El número de elementos que están en el intervalo $[-5, 10)$

c) $\|A\| = \sqrt{\sum_{i=1}^n \sum_{j=1}^n a_{ij}^2}$

[47] Escriba un programa en C que solicite al usuario el orden n y los elementos a_{ij} enteros de una matriz cuadrada, y además

a) muestre por pantalla un arreglo que contenga todos los números pares.

b) calcule la norma euclidiana del arreglo obtenido en a) y la muestre por pantalla usando color rojo.

c) muestre por consola la suma de cada uno de los elementos del arreglo obtenido en a) elevados al cubo.

[48] Escriba un programa en C, que use el comando switch para permitirle al usuario elegir calcular el determinante o la traza de una matriz de orden 2, cuyos elementos son ingresados por el usuario.

[49] Escriba un programa en C que, haciendo uso de la orden switch, muestre un menú que permita calcular la traza de una matriz cuadrada de orden 2 o los valores propios de una matriz de orden 2, cuyos elementos los ingresa el usuario

[50] Escriba un programa en C que muestre por pantalla, un menú que permita al usuario elegir entre calcular el determinante, la inversa o los valores propios de una matriz $A = (a_{ij}) \in \mathcal{M}_2(\mathbb{R})$ cuyos elementos son ingresados por el usuario. El programa debe cubrir todos los casos posibles.

[51] Escriba un programa en C que muestre por pantalla la inversa de una matriz $A = (a_{ij}) \in \mathcal{M}_2(\mathbb{Z})$ cuyos elementos, números entre -3 y 35 inclusive, son generados aleatoriamente. El programa debe considerar todos los casos posibles.

[52] Escriba un programa en C que solicite al usuario el ingreso de los elementos de una matriz $A \in \mathcal{M}_2(\mathbb{R})$ y muestre por pantalla el resultado de $(A^2)^{-1}$

[53] Escriba una aplicación en C, que permita calcular la inversa de una matriz triangular de orden 3. El usuario elige si va a ingresar una matriz triangular superior o inferior.

[54] Escriba un programa en C que muestre por pantalla la inversa de una matriz de orden 2, cuyos coeficientes los ingresa el usuario. Implemente una función que calcule la inversa de una matriz de orden 2.

[55] Escriba las siguientes funciones :

- a) MTraspuesta : acepta una matriz como argumento y entrega como resultado la traspuesta.
- b) MSimetrica : acepta una matriz como argumento y devuelve 1 si dicha matriz es simétrica y 0 si no lo es.

[56] Escriba un programa en C que calcule y muestre por pantalla la inversa de una matriz triangular superior de orden 3 ingresada por el usuario. Indicación: Sea $B = (b_{ij})$ la inversa de una matriz triangular superior $A = (a_{ij})$. Luego :

$$b_{ii} = \frac{1}{a_{ii}} ; i = 1, 2, \dots, n$$

$$b_{i,j} = \frac{-\sum_{k=i+1}^j a_{i,k} b_{k,j}}{a_{i,i}} ; j = n, n-1, \dots, 1; i = j-1, j-2, \dots, 1$$

[57] Escriba un programa en C que muestre por pantalla la norma del máximo $\|A\|_{\text{maximo}} = \max_{1 \leq i \leq 3} \left\{ \sum_{j=1}^3 |a_{i,j}| \right\}$ de una matriz cuadrada de orden 3 con elementos reales.

Reescriba el programa anterior para $\|A\|_{\text{maximo}} = \max_{1 \leq j \leq 3} \left\{ \sum_{i=1}^3 |a_{i,j}| \right\}$

[58] Escriba un programa en C que calcule y muestre por pantalla la norma espectral de una matriz cuadrada de orden 2, con elementos reales ingresados por el usuario.

(Indicación: La norma espectral de A es $\|A\| = \sqrt{\lambda_{\text{maximo}}(A^T A)}$, donde λ_{maximo} es el mayor valor propio de la matriz $A^T A$)

[59] Escriba un programa en C que calcule y muestre por pantalla la norma de una matriz $A \in \mathcal{M}_{m \times n}(\mathbb{R})$, donde el orden de la matriz y los elementos de ésta son ingresados por el usuario.

(Indicación: $\|A\| = \sqrt{\text{Traza}(A^T \cdot A)}$)

[60] Escriba un programa en C que muestre por pantalla la solución del sistema

$$\begin{aligned} a_{11}x + a_{12}y &= b_1 \\ a_{21}x + a_{22}y &= b_2 \end{aligned}$$

usando el método de Cramer.

El usuario es quien ingresa los coeficientes $a_{i,j}$; $i, j = 1, 2$, y b_i ; $i = 1, 2$.

¿Qué modificaciones le haría al programa si el sistema es de orden 3?

[61] Use C para resolver un sistema de ecuaciones de orden 2 por el método de la inversa. El usuario es quien ingresa los elementos de la matriz de coeficientes y el vector de constantes.
El programa debe emitir un mensaje apropiado cuando el determinante de la matriz de coeficientes sea cero.

[62] a) Escriba un programa en C que resuelva un sistema de ecuaciones lineales de orden 2 mediante el método de Cramer, haciendo uso de una función que calcule el determinante de una matriz de orden 2 definida por usted.
b) Modifique el programa anterior de modo que resuelva un sistema de orden 3.

[63] Use C para resolver el sistema

$$\begin{aligned}ax + by &= c \\dx + ey &= f\end{aligned}$$

donde a, b, c, d, e y f son constantes reales ingresadas por el usuario.
Considere todos los casos posibles.

[64] Escriba un programa en C que determine si una matriz cuadrada de orden n es antisimétrica o no. El usuario es quien ingresa el orden de la matriz y sus elementos.

[65] Escriba una función en C que determine si una matriz cuadrada de orden 3 es antisimétrica o no.
Use la función anterior para determinar la norma euclidiana de una matriz antisimétrica ingresada por el usuario.

[66] Una matriz $A \in \mathcal{M}_n(\mathbb{R})$ se dice **bien condicionada** si $k(A)$ es mayor o igual que 0.95 y menor o igual que 1.05, donde $k(A) = \|A\| \|A^{-1}\|$. Use C para verificar si una matriz cuadrada de orden 2, cuyos elementos son ingresados por el usuario, está bien condicionada. Trabaje con la norma euclidiana.

[67] Escriba un programa en C, que muestre por pantalla el polinomio de Lagrange evaluado en $x = c$, considerando los puntos (x_i, y_i) , $i = 0, 1, 2, \dots, n$, donde el valor de n , los puntos, y el valor de c son ingresados por el usuario. Solicite al usuario el ingreso obligatorio de $c \in (x_0, x_n)$. (Indicación: El polinomio de Lagrange es

$$L(x) = \sum_{i=0}^n y_i l_i(x)$$

$$\text{con } l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x-x_j}{x_i-x_j}$$

[68] Escriba un programa en C que solicite al usuario el ingreso de las estaturas de n alumnos, cuyas alturas no deben superar los 2.25 metros, para luego calcular y mostrar por pantalla la estatura promedio, y si tal promedio es mayor que 1.58, entonces calcule la desviación estándar. El valor de n es aleatorio y menor o igual que 12. El proceso anterior completo, debe repetirse cinco veces.

[69] Escriba un programa en C que calcule y muestre por pantalla, con cuatro decimales, el producto interior $\langle A, B \rangle$, con $A \in \mathcal{M}_{m \times n}(\mathbb{R})$ y $B \in \mathcal{M}_{m \times n}(\mathbb{R})$. El usuario es quien ingresa el orden y los elementos de cada matriz. El programa además, debe mostrar por pantalla un mensaje que indique si las matrices son ortogonales o no.

[70] Usando la orden switch, escriba un programa en C que muestre por pantalla un menú que permita al usuario elegir entre las opciones a) o b) siguientes:

a) Mostrar por pantalla la traza de una matriz de orden m cuyos elementos, en el intervalo con 1 decimal $[-15.2; 14.6]$, son generados aleatoriamente. El valor de m es un número entero mayor que 2 ingresado por el usuario. Debe utilizar una función de nombre traza creada por usted.

b) Muestre por pantalla los primeros 20 polinomios de Laguerre evaluados en 8, y de lo anterior calcule además la suma de los números menores que 10, y finalmente el producto de los números mayores o iguales que 12.

(Ind. : $P_0(x) = 1$; $P_1(x) = 1 - x$; $P_{n+1}(x) = \frac{(2n+1-x)}{n+1} P_n(x) - \frac{n}{n+1} P_{n-1}(x)$)

[71] Escriba una aplicación de consola, usando matrices y vectores, que calcule y muestre por pantalla, en formato matricial, el producto $A^{-1} \cdot b$, con $A = \begin{cases} 2+i & , i=j \\ j-2 & , i \neq j \end{cases}$ y $b = \begin{bmatrix} 1 \\ -5 \end{bmatrix}$.

Considere que la matriz A es de orden 2, y que el programa debe calcular todo, incluida la inversa.

[72] Escriba un programa en C que muestre por pantalla el resultado de $\frac{h}{3} \left[f(x_0) + 4 \sum_{i=1}^{\frac{n}{2}} f(x_{2i-1}) + 2 \sum_{i=1}^{\frac{n-2}{2}} f(x_{2i}) + f(x_n) \right]$, con $f(x) = x^3 - 1$ y $h = \frac{b-a}{n}$

El usuario ingresa los valores de a y b , con $a < b$

El n es un número aleatorio par, perteneciente al intervalo $[2, 30]$

Además $x_i = a + h i$, para $i = 0, 1, 2, \dots, n$

Note que: $x_0 = a$ y $x_n = b$

[73] Escriba una aplicación en lenguaje C que muestre por pantalla, en formato matricial, la traspuesta de una matriz de orden $m \times n$ cuyos elementos son ingresados por el usuario, al igual que el orden. El programa debe incluir y utilizar una función que trasponga una matriz.

PROBLEMAS RESUELTOS TIPOS ESTRUCTURADOS

3

Escriba un programa en C que calcule el promedio y la desviación estándar de n notas del 1 al 7, donde n es un número natural ingresado por el usuario, al igual que las notas.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

/*Escriba un programa en C que calcule el promedio
y la desviacion estandar de n notas del 1 al 7,
donde n es un numero natural ingresado por el usuario,
al igual que las notas. */

int main()
{
    float nota[1000];
    int n, i;
    float suma, suma2=0.0, Promedio, DesvEst;

    printf("\nIngrese el numero de notas :");
    scanf("%d",&n);

    for (i=0;i<=n-1;i++)
    {
        printf("\nNota(%i) = ",i+1);
        scanf("%f",&nota[i]);
```

```

}
suma=0.0;
for (i=0;i<=n-1;i++)
    suma=suma+nota[i];

Promedio=suma/n;

for (i=0;i<=n-1;i++)
    suma2=suma2+pow(Promedio-nota[i],2);

DesvEst=sqrt(suma2/(n-1));

printf("\nEl promedio es %0.1f\n\n", Promedio);
printf("\nLa desviacion estandar es %0.1f\n\n", DesvEst);

system("PAUSE");
return 0;
}

```

Salida

```

Ingrese el numero de notas :4
Nota<1> = 2.1
Nota<2> = 3.4
Nota<3> = 5.7
Nota<4> = 6.8
El promedio es 4.5

La desviacion estandar es 2.1
Presione una tecla para continuar . . .

```

Observaciones:

1) En este ejercicio no se han incluido los tildes

2) El promedio de n notas x_i se calcula usando la fórmula $\text{Promedio} = \frac{\sum_{i=1}^n x_i}{n}$

3) La desviación estándar se calcula usando la fórmula

$$\text{DevEst} = \sqrt{\frac{\sum_{i=1}^n (\text{Promedio} - x_i)^2}{n-1}}$$

4) Si el usuario ingresa un número de notas igual a 1, se produce una división por cero, que queda reflejada en la siguiente salida

```
Ingrese el numero de notas :1
Nota<1> = 3.4
El promedio es 3.4
La desviacion estandar es -1.5
Presione una tecla para continuar . . .
```

5) La línea `float nota[1000];` reserva 1000 lugares de memoria para almacenar hasta 1000 notas. Hacer esto no es eficiente porque se reservan muchos lugares de memoria para quizás un número reducido de notas (por ejemplo 4). Una mejor manera de trabajar esto es, luego del ingreso del número de notas por parte del usuario, reservar las que realmente se necesitan, como se incluye en el siguiente código fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
```

```
/*Escriba un programa en C que calcule el promedio
y la desviacion estandar de n notas del 1 al 7,
donde n es un numero natural ingresado por el usuario,
al igual que las notas. */
```

```
int main()
{
    int n, i;
    float suma, suma2=0.0, Promedio, DesvEst;

    printf("\nIngrese el numero de notas :");
    scanf("%d",&n);

    float nota[n];
```

```

for (i=0;i<=n-1;i++)
{
    printf("\nNota(%i) = ",i+1);
    scanf("%f",&nota[i]);
}
suma=0.0;
for (i=0;i<=n-1;i++)
    suma=suma+nota[i];

Promedio=suma/n;

for (i=0;i<=n-1;i++)
    suma2=suma2+pow(Promedio-nota[i],2);

DesvEst=sqrt(suma2/(n-1));

printf("\nEl promedio es %0.1f\n\n", Promedio);
printf("\nLa desviacion estandar es %0.1f\n\n", DesvEst);

system("PAUSE");
return 0;
}

```

cuya salida es

```

Ingrese el numero de notas :4
Nota<1> = 3.2
Nota<2> = 4.6
Nota<3> = 3.2
Nota<4> = 6.4
El promedio es 4.3
La desviacion estandar es 1.5
Presione una tecla para continuar . . .

```

5

Use C para determinar si un vector de \mathbb{R}^n es unitario o no. Las componentes del vector son ingresadas por el usuario, con n un número natural aleatorio no mayor a 20.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>

int main()
{
    srand(time(NULL));
    int n, i;
    n=rand()%20+1;
    float v[n], suma=0.0, norma;

    for(i=1;i<=n;i++)
    {
        printf("\nv[%i] = ",i);
        scanf("%f",&v[i]);
    }
    for(i=1;i<=n;i++)
        suma=suma+v[i]*v[i];
    norma=sqrt(suma);

    if(norma==1.0) printf("\nEl vector es unitario\n");
    else printf("\nEl vector no es unitario\n");

    system("PAUSE");
    return 0;
}
```

Salida

```
v[1] = -1
v[2] = 0
v[3] = 2
El vector no es unitario
Presione una tecla para continuar . . .
```

Observaciones:

1) En este ejercicio es preciso generar primero el valor de n y luego declarar el vector v , cuya dimensión depende de n .

2) La norma de $v = [v_1, v_2, \dots, v_n]$ es $\|v\| = \sqrt{v_1^2 + v_2^2 + \dots + v_n^2}$

6

Escriba un programa en C que muestre por pantalla el producto interior de dos vectores de 15 componentes enteras ingresadas por el usuario e indique si los vectores son ortogonales o no.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    const n=15;
    int v1[n], v2[n];
    int suma=0;
    int i;
```



```
for (i=1;i<=n;i++)
{
    printf("\nIngrese v1(%d)= ",i);
    scanf("%d",&v1[i]);
    printf("\nIngrese v2(%d)= ",i);
    scanf("%d",&v2[i]);
}
for (i=1;i<=n;i++)
    suma=suma+v1[i]*v2[i];

printf("\nEl producto interior es %d\n\n", suma);
if (suma==0) printf("\nLos vectores son ortogonales\n\n");
else printf("\nLos vectores no son ortogonales\n\n");

system("PAUSE");
return 0;
}
```

Salida

Ingrese $v_1(1) = 1$

Ingrese $v_2(1) = 2$

Ingrese $v_1(2) = 3$

Ingrese $v_2(2) = 4$

Ingrese $v_1(3) = 5$

Ingrese $v_2(3) = 6$

Ingrese $v_1(4) = 7$

Ingrese $v_2(4) = 8$

Ingrese $v_1(5) = 8$

Ingrese $v_2(5) = 9$

Ingrese $v_1(6) = 0$

Ingrese $v_2(6) = 1$

Ingrese $v_1(7) = 2$

Ingrese $v_2(7) = 3$

Ingrese $v_1(8) = 4$

Ingrese $v_2(8) = 5$

Ingrese $v_1(9) = 6$

Ingrese $v_2(9) = 7$

Ingrese $v_1(10) = 8$

Ingrese $v_2(10) = 9$

Ingrese $v_1(11) = 1$

Ingrese $v_2(11) = 0$

Ingrese $v_1(12) = 0$

Ingrese $v_2(12) = 0$

Ingrese $v_1(13) = 0$

Ingrese $v_2(13) = 0$

Ingrese $v_1(14) = 0$

Ingrese $v_2(14) = 0$

Ingrese $v_1(15) = 0$

Ingrese $v_2(15) = 0$

El producto interior es 312

Los vectores no son ortogonales

Presione una tecla para continuar . . .

Observaciones:

1) Si $\mathbf{a} = [a_1, a_2, \dots, a_n]$ y $\mathbf{b} = [b_1, b_2, \dots, b_n]$ son dos vectores de \mathbb{R}^n , entonces el producto interior entre ellos es $\langle \mathbf{a}, \mathbf{b} \rangle = \sum_{i=1}^n a_i b_i$

2) La variable suma almacena el producto interior entre los vectores ingresados por el usuario.

3) Dado que se ha utilizado sólo un bucle for para el ingreso de ambos vectores, las componentes de éstos aparecen intercaladas. \square

7

Usando vectores, calcule y muestre por pantalla, en grados, el ángulo entre los vectores $\mathbf{a} = [a_1, a_2, a_3]$ y $\mathbf{b} = [b_1, b_2, b_3]$, donde \mathbf{a} y \mathbf{b} son ingresados por el usuario.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    int n=3, i;
    float a[n],b[n];
    float ProdPunto=0, AnguloRad, AnguloGrad, NormaA=0, NormaB=0;
    // Ingreso de las componentes de los vectores a y b
    for(i=0;i<=n-1;i++)
    {
        printf("\nIngrese a[%i]=" ,i+1);
        scanf("%f",&a[i]);
    }
}
```

```

}
for(i=0;i<=n-1;i++)
{
    printf("\nIngrese b[%i]=",i+1);
    scanf("%f",&b[i]);
}
// Cálculo del producto interior
for(i=0;i<=n-1;i++)
    ProdPunto=ProdPunto+a[i]*b[i];
// Cálculo de la norma de a
for(i=0;i<=n-1;i++)
    NormaA=NormaA+pow(a[i],2);
NormaA=sqrt(NormaA);
//Cálculo de la norma de b
for(i=0;i<=n-1;i++)
    NormaB=NormaB+pow(b[i],2);
NormaB=sqrt(NormaB);
// Cálculo del ángulo entre los vectores en radianes
AnguloRad=acos(ProdPunto/(NormaA*NormaB));
// Conversión del ángulo en radianes a grados
AnguloGrad=180/M_PI*AnguloRad;
// Se muestra el angulo en grados por pantalla
printf("\nEl %cngulo entre los vectores es %0.2f grados\n\n",160,AnguloGrad);
system("PAUSE");
return 0;
}

```

Salida 1

```

Ingrese a[1]=1
Ingrese a[2]=1
Ingrese a[3]=1
Ingrese b[1]=-1
Ingrese b[2]=1
Ingrese b[3]=0
El ángulo entre los vectores es 90.00 grados
Presione una tecla para continuar . . .

```

Salida 2

```
Ingrese a[1]=1
Ingrese a[2]=1
Ingrese a[3]=1
Ingrese b[1]=1
Ingrese b[2]=1
Ingrese b[3]=1
El ángulo entre los vectores es -1.#J grados
Presione una tecla para continuar . . .
```

Observaciones:

1) Si $\mathbf{a} = [a_1, a_2, \dots, a_n]$ y $\mathbf{b} = [b_1, b_2, \dots, b_n]$ son dos vectores de \mathbb{R}^n , entonces el producto punto entre ellos es $\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^n a_i b_i$, la norma de \mathbf{a} es $\|\mathbf{a}\| = \sqrt{\sum_{i=1}^n a_i^2}$

y la norma de \mathbf{b} es $\|\mathbf{b}\| = \sqrt{\sum_{i=1}^n b_i^2}$

2) En la salida 2 se observa que el ángulo entre los vectores es -1.#J, que muestra que existe algún error en el cálculo. Lo que sucede aquí es que debido a las aproximaciones que se realizan sobre números reales el argumento de *Arccoseno* (acos) es un número mayor que 1, y no exactamente igual a 1, lo que produce el error. El resultado correcto en esta salida es 0 grados, pues los vectores son iguales.

3) En el código fuente se han incluido comentarios en línea para describir lo que hace cada sección de código. El incluir comentarios es una práctica muy recomendada en proyectos de muchas líneas de código y también cuando trabaja más de una persona en un proyecto.

8

Escriba un programa en C que calcule la norma euclidiana del vector $\begin{bmatrix} 1 \\ -2 \\ 5 \end{bmatrix}$

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    int a[3]={1,-2,5}, i;
    float NormaEuclidiana=0.0;
    for(i=0;i<=2;i++)
        NormaEuclidiana=NormaEuclidiana+a[i]*a[i];
    NormaEuclidiana=sqrt(NormaEuclidiana);
    printf("\nLa norma euclidiana del vector [1,-2,5] es %f\n\n",NormaEuclidiana);
    system("PAUSE");
    return 0;
}
```

Salida

```
La norma euclidiana del vector [1,-2,5] es 5.477226
Presione una tecla para continuar . . .
```

Observaciones:

1) En la línea de código `int a[3]={1,-2,5}, i;` se inicializa el vector con las componentes indicadas en el enunciado.

2) El identificador `NormaEuclidiana` se utiliza en este programa para acumular la suma de cada componente al cuadrado y luego para calcular la raíz cuadrada de esta suma. □

9

Escriba un programa en C que calcule la distancia de la recta

$$L : \begin{cases} x = x_0 + at \\ y = y_0 + bt \\ z = z_0 + ct \end{cases}$$

al punto $P(1, -2, 0)$, donde (x_0, y_0, z_0) es ingresado por el usuario al igual que el vector director $[a, b, c] \neq [0, 0, 0]$

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    float p[3]={1,-2,0}, r[3], p0[3], pp0[3], ProdCruz[3];
    float NormaProdCruz=0.0, NormaVectorDir=0.0, Distancia;
    int i=0;
    for(i=0;i<=2;i++)
    {
        printf("\nIngrese p0[%i]=",i+1);
        scanf("%f",&p0[i]);
    }
    do
    {
        for(i=0;i<=2;i++)
        {
            printf("\nIngrese r[%i]=",i+1);
            scanf("%f",&r[i]);
        }
    }
    while(r[0]==0 && r[1]==0 && r[2]==0);
```

```

for(i=0;i<=2;i++)
  pp0[i]=p[i]-p0[i];
ProdCruz[0]=r[1]*pp0[2]-r[2]*pp0[1];
ProdCruz[1]=r[2]*pp0[0]-r[0]*pp0[2];
ProdCruz[2]=r[0]*pp0[1]-r[1]*pp0[0];
for(i=0;i<=2;i++)
  NormaProdCruz=NormaProdCruz+ProdCruz[i]*ProdCruz[i];
NormaProdCruz=sqrt(NormaProdCruz);
for(i=0;i<=2;i++)
  NormaVectorDir=NormaVectorDir+r[i]*r[i];
NormaVectorDir=sqrt(NormaVectorDir);
Distancia=NormaProdCruz/NormaVectorDir;
printf("\nLa distancia es %f\n\n",Distancia);
system("PAUSE");
return 0;
}

```

Salida 1

```

Ingrese p0[1]=1
Ingrese p0[2]=2
Ingrese p0[3]=3
Ingrese r[1]=2
Ingrese r[2]=3
Ingrese r[3]=1
La distancia es 2.988071
Presione una tecla para continuar . . .

```


Salida 2

```
Ingrese p0[1]=2
Ingrese p0[2]=1
Ingrese p0[3]=-1
Ingrese r[1]=0
Ingrese r[2]=0
Ingrese r[3]=0
Ingrese r[1]=0
Ingrese r[2]=0
Ingrese r[3]=0
Ingrese r[1]=2
Ingrese r[2]=1
Ingrese r[3]=3
La distancia es 3.273268
Presione una tecla para continuar . . .
```

Observaciones:

1) Sea L una recta con vector director $\mathbf{r} = [a, b, c]$ y sea $P = (p_1, p_2, p_3)$ un punto del espacio que no pertenece a la recta L . La distancia de P a L está dada por:

$$d(P, L) = \frac{\|\mathbf{r} \times \overrightarrow{PP_0}\|}{\|\mathbf{r}\|}$$

donde $P_0 = (x_0, y_0, z_0)$ es un punto cualquiera de la recta.

Notemos que el vector $\overrightarrow{PP_0} = [p_1 - x_0, p_2 - y_0, p_3 - z_0]$

$$\text{Además, } \mathbf{r} \times \overrightarrow{PP_0} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ a & b & c \\ p_1 - x_0 & p_2 - y_0 & p_3 - z_0 \end{vmatrix} =$$
$$[b(p_3 - z_0) - c(p_2 - y_0), c(p_1 - x_0) - a(p_3 - z_0), a(p_2 - y_0) - b(p_1 - x_0)]$$

2) En la salida 2 se muestra qué sucede si el usuario insiste en ingresar un vector director nulo; en este caso se repite el ingreso de las componentes del vector director hasta que se ingrese un vector no nulo. \square

10

Escriba un programa en C que solicite al usuario el ingreso las componentes de un vector $[v_1, v_2] \in \mathbb{R}^2$ y que muestre por pantalla, con 2 decimales, la norma del máximo. Además, el programa debe mostrar un mensaje en pantalla indicando cuando la norma es mayor o igual que 1.

(Obs.: La norma del máximo es : $\| [v_1, v_2] \| = \text{maximo}\{|v_1|, |v_2|\}$)

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    int n=2, i;
    float v[n], v0, v1, NormaMax=0;
    for(i=0;i<=n-1;i++)
    {
        printf("\nIngrese v[%i]=",i+1);
        scanf("%f",&v[i]);
    }
    v0=fabs(v[0]); v1=fabs(v[1]);
    if(v0>=v1) NormaMax=v0;
    else NormaMax=v1;
    printf("\nLa norma del m%cximo es %.2f\n",160,NormaMax);
    if(NormaMax>=1)
        printf("\nLa norma del m%cximo es mayor o igual que 1\n\n",160);
    system("PAUSE");
    return 0;
}
```

Salida 1

```
Ingrese v[1]=0.379
Ingrese v[2]=0.327
La norma del máximo es 0.38
Presione una tecla para continuar . . .
```

Salida 2

```
Ingrese v[1]=23.67
Ingrese v[2]=-34.68
La norma del máximo es 34.68
La norma del máximo es mayor o igual que 1
Presione una tecla para continuar . . .
```

Observaciones:

- 1) Se incluyen en el programa variables v_0 y v_1 para almacenar los valores absolutos de cada una de las componentes del vector v
- 2) En la salida 2 se ingresan componentes mayores que 1 para visualizar el mensaje correspondiente solicitado.

12

Escriba un programa en C que muestre por pantalla el producto escalar triple $\langle a, b \times c \rangle$, donde las componentes de los vectores son ingresadas por el usuario.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    float a[3], b[3], c[3];
    int i;
    float Prod1, Prod2, Prod3, ProdEscalarTriple;
    printf("\nIngrese las componentes del vector a\n");
    for(i=1;i<=3;i++)
    {
        printf("\n a(%i) = ",i);
        scanf("%f",&a[i]);
    }
    printf("\nIngrese las componentes del vector b\n");
    for(i=1;i<=3;i++)
    {
        printf("\n b(%i) = ",i);
        scanf("%f",&b[i]);
    }
    printf("\nIngrese las componentes del vector c\n");
    for(i=1;i<=3;i++)
    {
        printf("\n c(%i) = ",i);
        scanf("%f",&c[i]);
    }
    Prod1=b[2]*c[3]-b[3]*c[2];
    Prod2=b[3]*c[1]-b[1]*c[3];
    Prod3=b[1]*c[2]-b[2]*c[1];
    ProdEscalarTriple=a[1]*Prod1+a[2]*Prod2+a[3]*Prod3;
    printf("\nEl producto escalar triple <a , b x c> es %0.2f\n",ProdEscalarTriple);

    system("PAUSE");
    return 0;
}
```

Salida

```
Ingrese las componentes del vector a
a<1> = 1
a<2> = -1
a<3> = 2
Ingrese las componentes del vector b
b<1> = 1
b<2> = 1
b<3> = 1
Ingrese las componentes del vector c
c<1> = 2
c<2> = -1
c<3> = 3
El producto escalar triple <a , b x c> es -1.00
Presione una tecla para continuar . . .
```

Observaciones:

1) Sean $\mathbf{a} = [a_1, a_2, a_3]$, $\mathbf{b} = [b_1, b_2, b_3]$ y $\mathbf{c} = [c_1, c_2, c_3]$

$$\mathbf{b} \times \mathbf{c} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{vmatrix} = (b_2c_3 - b_3c_2)\mathbf{i} + (b_3c_1 - b_1c_3)\mathbf{j} + (b_1c_2 - b_2c_1)\mathbf{k} =$$

$$[b_2c_3 - b_3c_2, b_3c_1 - b_1c_3, b_1c_2 - b_2c_1]$$

Luego el producto vectorial triple, en términos de las componentes de los vectores participantes, es:

$$\langle \mathbf{a}, \mathbf{b} \times \mathbf{c} \rangle = \langle [a_1, a_2, a_3], [b_2c_3 - b_3c_2, b_3c_1 - b_1c_3, b_1c_2 - b_2c_1] \rangle =$$

$$a_1(b_2c_3 - b_3c_2) + a_2(b_3c_1 - b_1c_3) + a_3(b_1c_2 - b_2c_1)$$

2) Las variables Prod1, Prod2, y Prod3 almacenan las componentes del vector $\mathbf{b} \times \mathbf{c}$

13

Sean \mathbf{a} y \mathbf{b} dos vectores no nulos de \mathbb{R}^n . Se llama proyección de \mathbf{b} sobre \mathbf{a} al vector:

$$\text{proy}_{\mathbf{a}} \mathbf{b} = \frac{\langle \mathbf{a}, \mathbf{b} \rangle}{\|\mathbf{a}\|^2} \mathbf{a}$$

Escriba un programa en C que muestre por pantalla la proyección de \mathbf{b} sobre \mathbf{a} , donde \mathbf{a} y \mathbf{b} son ingresados por el usuario al igual que el valor de n .

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int n;
    printf("\nIngrese n : ");
    scanf("%i",&n);
    float a[n], b[n];
    int i;
    int Nuloa=0, Nulob=0;
    float ProdInt=0.0, suma=0.0, normaa;

    printf("\nIngrese las componentes del vector a\n");
    do
    {
        for(i=1;i<=n;i++)
        {
            printf("\n a(%i) = ",i);
            scanf("%f",&a[i]);
            if(a[i]!=0.0) Nuloa=1;
        }
    }
```

```

}
while(Nuloa==0);

printf("\nIngrese las componentes del vector b\n");
do
{
for(i=1;i<=n;i++)
{
printf("\n b(%i) = ",i);
scanf("%f",&b[i]);
if(b[i]!=0.0) Nulob=1;
}
}
while(Nulob==0);

for(i=1;i<=n;i++)
ProdInt=ProdInt+a[i]*b[i];

for(i=1;i<=n;i++)
suma=suma+a[i]*a[i];
normaa=suma;

for(i=1;i<=n;i++)
printf("\nproy(%i) = %0.2f\n",i,ProdInt/normaa*a[i]);

system("PAUSE");
return 0;
}

```

Salida 1

```
Ingrese n : 3
Ingrese las componentes del vector a
a<1> = 1
a<2> = 2
a<3> = 3
Ingrese las componentes del vector b
b<1> = -1
b<2> = 1
b<3> = -1
proy<1> = -0.14
proy<2> = -0.29
proy<3> = -0.43
Presione una tecla para continuar . . .
```

Salida 2

```
Ingrese n : 2
Ingrese las componentes del vector a
a<1> = 0
a<2> = 0
a<1> = 1
a<2> = -1
Ingrese las componentes del vector b
b<1> = 0
b<2> = 0
b<1> = 0
b<2> = 1
proy<1> = -0.50
proy<2> = 0.50
Presione una tecla para continuar . . .
```


Observaciones:

1) En este ejercicio se usan dos variables banderín (flag), que nos permiten detectar si el usuario ha ingresado un vector nulo; en tal caso se le solicita al usuario el reingreso del vector.

2) La variable ProdInt almacena el producto interior $\langle \mathbf{a}, \mathbf{b} \rangle$ y la variable norma almacena $\|\mathbf{a}\|^2$

14

Use el lenguaje de programación C para verificar si el conjunto $B = \{ [a_1, a_2, a_3], [b_1, b_2, b_3], [c_1, c_2, c_3] \}$ es ortonormal o no, donde las componentes de los vectores del conjunto son ingresadas por el usuario.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    int n=3, i, Ortonormal=1;
    float a[n], b[n], c[n];
    float ProdPuntoAB=0, ProdPuntoBC=0, ProdPuntoAC=0;
    float NormaA=0, NormaB=0, NormaC=0;
    for(i=0;i<=n-1;i++)
    {
        printf("\nIngrese a[%i] = ",i+1);
        scanf("%f",&a[i]);
    }
}
```

```

for(i=0;i<=n-1;i++)
{
printf("\nIngrese b[%i] = ",i+1);
scanf("%f",&b[i]);
}
for(i=0;i<=n-1;i++)
{
printf("\nIngrese c[%i] = ",i+1);
scanf("%f",&c[i]);
}
for(i=0;i<=n-1;i++)
ProdPuntoAB=ProdPuntoAB+a[i]*b[i];
if(ProdPuntoAB!=0.0) Ortonormal=0;
for(i=0;i<=n-1;i++)
ProdPuntoBC=ProdPuntoBC+b[i]*c[i];
if(ProdPuntoBC!=0.0) Ortonormal=0;
for(i=0;i<=n-1;i++)
ProdPuntoAC=ProdPuntoAC+a[i]*c[i];
if(ProdPuntoAC!=0.0) Ortonormal=0;
for(i=0;i<=n-1;i++)
NormaA=NormaA+a[i]*a[i];
NormaA=sqrt(NormaA);
if(NormaA!=1.0) Ortonormal=0;
for(i=0;i<=n-1;i++)
NormaB=NormaB+b[i]*b[i];
NormaB=sqrt(NormaB);
if(NormaB!=1.0) Ortonormal=0;
for(i=0;i<=n-1;i++)
NormaC=NormaC+c[i]*c[i];
NormaC=sqrt(NormaC);
if(NormaC!=1.0) Ortonormal=0;
if(Ortonormal==1) printf("\nEl conjunto ingresado es ortonormal\n\n");
else printf("\nEl conjunto ingresado no es ortonormal\n\n");
system("PAUSE");
return 0;
}

```

Salida 1

```
Ingrese a[1] = 1
Ingrese a[2] = 0
Ingrese a[3] = 0
Ingrese b[1] = 0
Ingrese b[2] = 1
Ingrese b[3] = 0
Ingrese c[1] = 0
Ingrese c[2] = 0
Ingrese c[3] = 1
El conjunto ingresado es ortonormal
Presione una tecla para continuar . . .
```

Salida 2

```
Ingrese a[1] = 1
Ingrese a[2] = -1
Ingrese a[3] = 0
Ingrese b[1] = 1
Ingrese b[2] = 1
Ingrese b[3] = 0
Ingrese c[1] = 0
Ingrese c[2] = 0
Ingrese c[3] = 1
El conjunto ingresado no es ortonormal
Presione una tecla para continuar . . .
```

Observaciones:

1) Un conjunto es ortonormal cuando es ortogonal y cada uno de sus vectores es unitario, es decir,

a) es ortogonal cuando

$$[a_1, a_2, a_3] \cdot [b_1, b_2, b_3] = 0$$

$$[b_1, b_2, b_3] \cdot [c_1, c_2, c_3] = 0$$

$$[a_1, a_2, a_3] \cdot [c_1, c_2, c_3] = 0$$

El producto punto es conmutativo, por lo que basta probar las tres condiciones anteriores para que el conjunto sea ortogonal.

b) es unitario cuando

$$\| [a_1, a_2, a_3] \| = 1$$

$$\| [b_1, b_2, b_3] \| = 1$$

$$\| [c_1, c_2, c_3] \| = 1$$

2) En el programa se ha utilizado una variable banderín (Ortonormal) que se inicializa en 1, que indica que el conjunto es ortonormal; a medida que alguna de las condiciones para que sea ortonormal no se cumple se le asigna el valor 0 a la variable Ortonormal.

3) En la salida 2 se ingresa un conjunto ortogonal, pero los vectores a y b no son unitarios, por lo que el conjunto no es ortonormal.

15

Escriba un programa en C que lea por pantalla el valor de n y las n componentes reales de un vector \mathbf{x} . Luego muestre por pantalla lo siguiente:

i) el vector \mathbf{x}

ii) la norma euclidiana de \mathbf{x}

iii) la suma de los elementos de índice impar

iv) un mensaje indicando si el promedio de los elementos de índice par es mayor o igual que 5.

Separe con una línea horizontal cada ítem anterior.

Obs.: Suponga que la primera componente de \mathbf{x} es $\mathbf{x}[1]$

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
```

```
int main()
```

```

{
  int n, i, k=0;
  printf("\nIngrese n = ");
  scanf("%i",&n);
  float x[n], NormaX=0, SumaIndImpar=0, SumaIndPar=0, Promedio;
  for(i=0;i<=n-1;i++)
  {
    printf("\nIngrese x[%i] = ",i+1);
    scanf("%f",&x[i]);
  }
  printf("_____ \n");
  printf("\nx = [");
  for(i=0;i<=n-1;i++)
  {
    printf(" %0.2f ",x[i]);
    if(i!=n-1) printf(",");
  }
  printf("]\n");
  printf("_____ \n");
  for(i=0;i<=n-1;i++)
    NormaX=NormaX+x[i]*x[i];
  NormaX=sqrt(NormaX);
  printf("\nLa norma euclidiana de x es %0.2f\n",NormaX);
  printf("_____ \n");
  for(i=0;i<=n-1;i=i+2)
    SumaIndImpar=SumaIndImpar+x[i];
  printf("\nLa suma de los elementos de %cndice impar es %0.2f\n",161,SumaIndImpar);
  printf("_____ \n");
  for(i=1;i<=n-1;i=i+2)
  {
    SumaIndPar=SumaIndPar+x[i];
    k=k+1;
  }
  Promedio=SumaIndPar/k;
  if(Promedio>=5)
  {
    printf("\nEl promedio de los elementos de %cndice par es mayor o igual que 5\n",161);
    printf("_____ \n");
  }
  printf("\n\n");
  system("PAUSE");
  return 0;
}

```

Salida 1

```
Ingrese n = 4
Ingrese x[1] = 5
Ingrese x[2] = 5
Ingrese x[3] = 5
Ingrese x[4] = 5

x = [ 5.00 , 5.00 , 5.00 , 5.00 ]

La norma euclidiana de x es 10.00

La suma de los elementos de índice impar es 10.00

El promedio de los elementos de índice par es mayor o igual que 5

Presione una tecla para continuar . . .
```

Salida 2

```
Ingrese n = 3
Ingrese x[1] = 1
Ingrese x[2] = 2
Ingrese x[3] = 3

x = [ 1.00 , 2.00 , 3.00 ]

La norma euclidiana de x es 3.74

La suma de los elementos de índice impar es 4.00

Presione una tecla para continuar . . .
```

Observaciones:

1) Para exhibir el vector x tal como aparece en la salida, es necesario mostrar primero el paréntesis cuadrado de apertura ([) y luego en un ciclo for mostrar las componentes del vector separadas por comas, excepto la última componente que no lleva coma. Terminado el ciclo cerramos el vector con]

2) En las sumas de los elementos de índice par e impar se ha utilizado la observación dada al final del enunciado, es decir, la primera componente del vector x es $x[1]$ y no $x[0]$ como internamente lo maneja el lenguaje C. Esto significa, por ejemplo, que las componentes de índice impar son $x[1], x[3], \dots$, tal como aparecen en la ejecución del programa.

3) En la salida 2 el promedio de los elementos de x de índice par no es mayor o igual que 5, por lo que no aparece mensaje.

16

Use C para determinar el mayor de n números ingresados por el usuario, donde n es un entero aleatorio en el intervalo $[1, 25]$.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    srand(time(NULL));
    int n=rand()%25+1;
    float v[n], mayor; int i;
    for(i=0;i<=n-1;i++)
    {
        printf("\nIngrese n%cmero %i = ",163,i+1);
        scanf("%f",&v[i]);
    }
    mayor=v[0];
    for(i=1;i<=n-1;i++)
        if(v[i]>mayor) mayor=v[i];
    else mayor=mayor;
    printf("\nEl mayor de los n%cmeros ingresados es %0.4f\n\n",163,mayor);
    system("PAUSE");
}
```

```
    return 0;  
}
```

Salida 1

```
Ingrese número 1 = 0.25  
Ingrese número 2 = 0.45  
Ingrese número 3 = 1.23  
El mayor de los números ingresados es 1.23  
Presione una tecla para continuar . . .
```

Salida 2

```
Ingrese número 1 = 7  
Ingrese número 2 = 7  
Ingrese número 3 = 7  
Ingrese número 4 = 7  
Ingrese número 5 = 7  
Ingrese número 6 = 7  
Ingrese número 7 = 7  
Ingrese número 8 = 7  
Ingrese número 9 = 7  
Ingrese número 10 = 7  
Ingrese número 11 = 7  
Ingrese número 12 = 7  
Ingrese número 13 = 7  
Ingrese número 14 = 7  
Ingrese número 15 = 7  
Ingrese número 16 = 7  
Ingrese número 17 = 7  
Ingrese número 18 = 7  
Ingrese número 19 = 7  
El mayor de los números ingresados es 7.00  
Presione una tecla para continuar . . .
```


Salida 3

```
Ingrese número 1 = 0.2121
Ingrese número 2 = 0.2123
El mayor de los números ingresados es 0.2123
Presione una tecla para continuar . . .
```

Observaciones:

1) Este ejercicio se trabajó con un arreglo, pues los arreglos permiten, entre otras cosas, almacenar y mantener los valores de los números ingresados para su posterior manejo.

2) El algoritmo utilizado en este ejercicio para obtener el mayor de un conjunto de números es el siguiente:

a) Hacer $\text{mayor} = v[0]$ (mayor se hace igual al primer número ingresado)
b) Si $v[i] > \text{mayor}$, entonces $\text{mayor} = v[i]$, en caso contrario $\text{mayor} = \text{mayor}$ (el valor de mayor sigue siendo el mismo, a menos que el número ingresado sea mayor, en cuyo caso se almacena este nuevo valor en la variable mayor. Esto se repite para cada uno de los números ingresados hasta terminar. La variable mayor almacena finalmente el valor mayor de todos los números ingresados)

3) Un posible algoritmo para determinar el menor de un conjunto de números es análogo al mencionado en 2)

4) Las salidas mostradas difieren en cantidad de números ingresados porque tal cantidad es un número aleatorio en el intervalo cerrado $[1, 25]$

Escriba un programa en C que genere en forma aleatoria m notas reales, del 1 al 7 con un decimal, para cada uno de los n alumnos de un curso, mostrando por pantalla los alumnos aprobados (promedio ≥ 4) junto con su promedio, usando dos decimales. Numere los alumnos del 1 al n para identificarlos. Los valores m y n son ingresados por el usuario, quien debe ser obligado a ingresar números naturales mayores o iguales que 4, en ambos casos. El programa además debe mostrar por pantalla el número de alumnos con distinción (promedio $\in [5, 5.6]$) y distinción máxima (promedio ≥ 5.7).

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    int m, n;
    do
    {
        printf("\nLos valores de m y n deben ser iguales o superiores a 4\n\n");
        printf("\nIngrese el número de notas m=",163);
        scanf("%i",&m);
        printf("\nIngrese el número de alumnos n=",163);
        scanf("%i",&n);
    }
    while (m<4 || n<4);

    float a[n][m], suma[n], promedio[n];
    int i, j, Distincion=0, DistincionMaxima=0;

    srand(time(NULL));

    for (i=1;i<=n;i++)
        for (j=1;j<=m;j++)
```

```

a[i][j]=(rand()%61+10)/10.;

for (i=1;i<=n;i++)
{ suma[i]=0;
  for (j=1;j<=m;j++)
    suma[i]=suma[i]+a[i][j];
}

for (i=1;i<=n;i++)
{
  promedio[i]=suma[i]/m;
  if (promedio[i]>=4.0)
  {
    printf("\nEl alumno %i est%c aprobado",i,160);
    printf(" con un promedio igual a %0.2f\n",promedio[i]);
  }
}
for (i=1;i<=n;i++)
{
  if (promedio[i]>=5 && promedio[i]<=5.6) Distincion++;
  else
  if (promedio[i]>=5.7 && promedio[i]<=7.0) DistincionMaxima++;
}

printf("\nEl n%cmero de alumnos con distincion es %i\n",163,162,Distincion);
printf("\nEl n%cmero de alumnos con distincion",163,162);
printf(" maxima es %i\n\n",160,DistincionMaxima);

printf("\n\n");
system("PAUSE");
return 0;
}

printf("\nEl n%cmero de alumnos con distincion es %i\n",163,162,Distincion);
printf("\nEl n%cmero de alumnos con distincion es %i\n",163,162,Distincion);
printf("\nEl n%cmero de alumnos con distincion",163,162);

system("PAUSE");
return 0;
}

```

Salida

```
Los valores de m y n deben ser iguales o superiores a 4
Ingrese el número de notas m=10
Ingrese el número de alumnos n=25
El alumno 3 está aprobado con un promedio igual a 5.10
El alumno 6 está aprobado con un promedio igual a 4.51
El alumno 9 está aprobado con un promedio igual a 4.52
El alumno 11 está aprobado con un promedio igual a 4.50
El alumno 12 está aprobado con un promedio igual a 4.38
El alumno 14 está aprobado con un promedio igual a 4.36
El alumno 15 está aprobado con un promedio igual a 4.38
El alumno 17 está aprobado con un promedio igual a 4.73
El alumno 22 está aprobado con un promedio igual a 4.32
El alumno 24 está aprobado con un promedio igual a 4.25
El alumno 25 está aprobado con un promedio igual a 4.34
El número de alumnos con distinción es 1
El número de alumnos con distinción máxima es 0

Presione una tecla para continuar . . .
```

Observaciones:

1) En este ejercicio se utiliza la matriz $a[i][j]$ para almacenar las notas y el número correspondiente a cada alumno. Las filas representan los números asociados a los alumnos y las columnas las notas. Esto significa por ejemplo, que $a[1][j]$ con $j = 1, 2, \dots, m$, representa las notas del alumno 1.

2) La variable $\text{promedio}[i]$ almacena el promedio del i -ésimo alumno.

El coeficiente de correlación nos entrega una medida de cómo dos variables aleatorias están asociadas.

Si $\vec{x} = (x_1, x_2, x_3, \dots, x_n)$ y $\vec{y} = (y_1, y_2, y_3, \dots, y_n)$ son vectores que almacenan los datos asociados a las variables, entonces el coeficiente de correlación está dado por la fórmula:

$$r = \frac{\sum_{i=1}^n x_i y_i - n \bar{x} \bar{y}}{\sqrt{\left(\sum_{i=1}^n x_i^2 - n (\bar{x})^2 \right) \cdot \left(\sum_{i=1}^n y_i^2 - n (\bar{y})^2 \right)}}$$

$$\text{con } \bar{x} = \frac{\sum_{i=1}^n x_i}{n} \text{ y } \bar{y} = \frac{\sum_{i=1}^n y_i}{n}$$

Escriba un programa en C que calcule y muestre por pantalla el coeficiente de correlación de dos vectores de n componentes reales generadas aleatoriamente, donde n es ingresado por el usuario. Suponga que las componentes de los vectores son números reales con un decimal, en el intervalo $[-100, 100]$.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>

int main()
{
    srand(time(NULL));
    int n, i;
    printf("\nIngrese n = ");
    scanf("%i",&n);
    float x[n], y[n];
```

```

for(i=0;i<=n-1;i++)
{
  x[i]=(rand()%2001-1000)/10.0;
  y[i]=(rand()%2001-1000)/10.0;
}
float Sumax=0, Sumay=0, Promx, Promy;
float Sumaxy=0, Sumax2=0, Sumay2=0;
float r;
for(i=0;i<=n-1;i++)
{
  Sumax=Sumax+x[i];
  Sumay=Sumay+y[i];
  Sumaxy=Sumaxy+x[i]*y[i];
  Sumax2=Sumax2+x[i]*x[i];
  Sumay2=Sumay2+y[i]*y[i];
}
Promx=Sumax/n; Promy=Sumay/n;
r=(Sumaxy-n*Promx*Promy)/sqrt((Sumax2-n*Promx*Promx)*(Sumay2-
n*Promy*Promy));
printf("\nEl coeficiente de correlaci%cn es %f\n",162,r);
printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida 1

```

Ingrese n = 1000
El coeficiente de correlación es -0.063162
Presione una tecla para continuar . . .

```

Salida 2

```
Ingrese n = 5
El coeficiente de correlación es 0.237515
Presione una tecla para continuar . . .
```

Salida 3

```
Ingrese n = 1
El coeficiente de correlación es -1.#IND00
Presione una tecla para continuar . . .
```

Observaciones:

- 1) El coeficiente de correlación es un número real en el intervalo cerrado $[-1, 1]$
- 2) En la salida 3 se observa que para $n = 1$ el resultado es $-1.\#IND00$, lo que indica que se ha producido algún tipo de error. Este error se produce porque la cantidad subradical puede llegar a ser negativa o cero cuando $n = 1$.

27

Escriba un programa en C que calcule y muestre por pantalla, en formato matricial, la suma $3A + 2B$, donde A y B son matrices cuadradas de orden 6 ingresadas por el usuario.
El programa además debe mostrar por pantalla la traza de $3A + 2B$.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
```

```

int main()
{
    const n=6;
    float a[n][n], b[n][n], suma[n][n];
    float traza=0.0;
    int i,j;

    for(i=1;i<=n;i++)
    for(j=1;j<=n;j++)
    {
        printf("\nIngrese a(%i,%i) = ",i,j);
        scanf("%f",&a[i][j]);
    }
    for(i=1;i<=n;i++)
    for(j=1;j<=n;j++)
    {
        printf("\nIngrese b(%i,%i) = ",i,j);
        scanf("%f",&b[i][j]);
    }
    for(i=1;i<=n;i++)
    for(j=1;j<=n;j++)
    {
        suma[i][j]=3*a[i][j]+2*b[i][j];
    }
    for(i=1;i<=n;i++)
    { printf("\n");
    for(j=1;j<=n;j++)
        printf("\t%0.1f ",suma[i][j]);
    }
    for(i=1;i<=n;i++)
        traza=traza + suma[i][i];
    printf("\n\nLa traza de 3A + 2B es %0.2f\n\n",traza);

    system("PAUSE");
    return 0;
}

```


Parte de la Salida

En esta salida se asignaron sólo números 1 a la matriz a y sólo números 2 a la matriz b.

```
Ingrese b<5,3> = 2
Ingrese b<5,4> = 2
Ingrese b<5,5> = 2
Ingrese b<5,6> = 2
Ingrese b<6,1> = 2
Ingrese b<6,2> = 2
Ingrese b<6,3> = 2
Ingrese b<6,4> = 2
Ingrese b<6,5> = 2
Ingrese b<6,6> = 2
      7.0    7.0    7.0    7.0    7.0    7.0
      7.0    7.0    7.0    7.0    7.0    7.0
      7.0    7.0    7.0    7.0    7.0    7.0
      7.0    7.0    7.0    7.0    7.0    7.0
      7.0    7.0    7.0    7.0    7.0    7.0
      7.0    7.0    7.0    7.0    7.0    7.0
La traza de 3A + 2B es 42.00
Presione una tecla para continuar . . .
```

Observaciones:

1) Dado que el orden de las matrices es 6 y no cambia durante el programa se escribe la declaración `const n=6;`

2) La variable suma almacena la suma $3A + 2B$

3) La sección de código

```
for(i=1;i<=n;i++)
{ printf("\n");
  for(j=1;j<=n;j++)
    printf("\t%.1f ",suma[i][j]);
}
```

permite mostrar suma en formato matricial, es decir, los elementos acomodados en filas y columnas. □

45

Escriba una aplicación en C, que calcule y muestre por pantalla la traza de una matriz cuadrada, cuyo orden es un número ingresado por el usuario, y cuyos elementos son números aleatorios impares en el intervalo $[-3, 13]$. El programa debe obligar al usuario a ingresar un orden mayor o igual que 2.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    int n;
    do
    {
        printf("\nIngrese del orden de la matriz : ");
        scanf("%i",&n);
    }
    while(n<2);

    int a[n][n], i, j, traza=0;

    srand(time(NULL));

    for(i=1;i<=n;i++)
        for(j=1;j<=n;j++)
            do
```

```

{
  a[i][j]=rand()%17-3;
}
while(a[i][j]%2==0);

for(i=1;i<=n;i++)
  traza=traza+a[i][i];

printf("\nLa traza es %i\n",traza);

printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida

```

Ingrese del orden de la matriz : 4
La traza es 18
Presione una tecla para continuar . . .

```

Observaciones:

1) La sección de código

```

do
{
  a[i][j]=rand()%17-3;
}
while(a[i][j]%2==0);

```

permite asignar a la matriz sólo elementos pares.

2) Dado que la matriz posee elementos aleatorios se declara de tipo int.

46

Escriba un programa en C que solicite al usuario el ingreso del orden y de los elementos de una matriz $A \in \mathcal{M}_n(\mathbb{R})$, y que muestre por pantalla :

a) La suma de los elementos extradiagonales.

b) El número de elementos que están en el intervalo $[-5, 10)$

$$c) \|A\| = \sqrt{\sum_{i=1}^n \sum_{j=1}^n a_{ij}^2}$$

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    int i,j, n;
    float suma=0.0, sumaT=0.0;
    int Cuenta=0;
    printf("Ingrese el orden de la matriz : ");
    scanf("%i",&n);
    float A[n][n];
    for (i=1;i<=n;i++)
        for (j=1;j<=n;j++)
        {
            printf("A(%i,%i)= ",i,j);
            scanf("%f",&A[i][j]);
        }
    for (i=1;i<=n;i++)
        for (j=1;j<=n;j++)
        {
            if (i!=j) suma=suma+A[i][j];
            if ((A[i][j]>=-5.0)&&(A[i][j]<10)) Cuenta=Cuenta+1;
            sumaT=sumaT+A[i][j]*A[i][j];
        }
}
```

```

printf("La suma de los elementos extradiagonales es %0.2f\n",suma);
printf("El número de elementos que están en el intervalo [-5,10) es %i\n",163,160,Cuenta);
printf("La norma de A es %0.2f\n",sqrt(sumaT));

printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida

```

Ingrese el orden de la matriz : 2
A(1,1)= 1
A(1,2)= 2
A(2,1)= 3
A(2,2)= 4
La suma de los elementos extradiagonales es 5.00
El número de elementos que están en el intervalo [-5,10) es 4
La norma de A es 5.48

Presione una tecla para continuar . . .

```

Observaciones:

- 1) Recordemos que los elementos extradiagonales de la matriz A son tales que $i \neq j$.
- 2) La variable Cuenta almacena el total de elementos que están en el intervalo $[-5, 10)$.

49

Escriba un programa en C que, haciendo uso de la orden switch, muestre un menú que permita calcular la traza de una matriz cuadrada de orden 2 o los valores propios de una matriz de orden 2, cuyos elementos los ingresa el usuario

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    const n=2;
    int opcion, i, j;
    float a[n][n], traza, discr, vp1, vp2, vp1r, vp1i, det;
    for(i=1;i<=n;i++)
        for(j=1;j<=n;j++)
        {
            printf("\na(%i,%i) = ",i,j);
            scanf("%f",&a[i][j]);
        }

    printf("\n1 Calcular la traza\n");
    printf("\n2 Calcular los valores propios\n");
    printf("\nIngrese opci%cn 1 %c 2 :",162,162);
    scanf("%i",&opcion);
    switch(opcion)
    {
        case 1:
            traza=a[1][1]+a[2][2];
            printf("\nLa traza es %0.1f\n",traza);
            break;
        case 2:
            traza=a[1][1]+a[2][2];
            det=a[1][1]*a[2][2]-a[1][2]*a[2][1];
            discr=pow(traza,2)-4*det;
            if(discr>0.0)
            {
                vp1=(traza+sqrt(discr))/2.;
                vp2=(traza-sqrt(discr))/2.;
                printf("\nLos valores propios son %0.2f y %0.2f\n",vp1,vp2);
            }
            else if(discr==0.0)
            {
```

```

    vp1=traza/2.;
    vp2=vp1;
    printf("\nLos valores propios son %0.2f y %0.2f\n",vp1,vp2);
}
else
{
    vp1r=traza/2.;
    vp1i=fabs(sqrt(-discr)/2.);
    printf("\nLos valores propios son %0.2f + %0.2f i\n",vp1r,vp1i);
    printf(" y %0.2f - %0.2f i\n",vp1r,vp1i);
}
break;
}
system("PAUSE");
return 0;
}

```

Salida 1

```

a(1,1) = 1
a(1,2) = -1
a(2,1) = 2
a(2,2) = 1
1 Calcular la traza
2 Calcular los valores propios
Ingrese opción 1 ó 2 :1
La traza es 2.0
Presione una tecla para continuar . . .

```

Salida 2

```
a<1,1> = -2
a<1,2> = 1
a<2,1> = 1
a<2,2> = 2
1 Calcular la traza
2 Calcular los valores propios
Ingrese opción 1 ó 2 :2
Los valores propios son 2.24 y -2.24
Presione una tecla para continuar . . .
```

Observaciones:

1) Se declara la variable n como const porque su valor no cambia durante el programa.

2) Notemos que si $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, entonces $\text{traza}(A) = a + d$ y los valores propios de A son

$$\lambda_1 = \frac{(a+d) + \sqrt{(a+d)^2 - 4(ad-bc)}}{2} = \frac{\text{traza}(A) + \sqrt{(\text{traza}(A))^2 - 4\text{Det}(A)}}{2}$$

y

$$\lambda_2 = \frac{(a+d) - \sqrt{(a+d)^2 - 4(ad-bc)}}{2} = \frac{\text{traza}(A) - \sqrt{(\text{traza}(A))^2 - 4\text{Det}(A)}}{2}$$

3) $\text{discr} = (\text{traza}(A))^2 - 4\text{Det}(A)$ \square

50

Escriba un programa en C que muestre por pantalla, un menú que permita al usuario elegir entre calcular el determinante, la inversa o los valores propios de una matriz $A = (a_{ij}) \in \mathcal{M}_2(\mathbb{R})$ cuyos elementos son ingresados por el usuario. El programa debe cubrir todos los casos posibles.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    const n=2;
    float a[n][n], ainv[n][n];
    float Determinante, Discriminante, ValPropio1, ValPropio2;
    float Temp, ParteReal, ParteImaginaria;
    unsigned int opcion; int i, j;

    for (i=1;i<=n;i++)
        for (j=1;j<=n;j++)
        {
            printf("\na(%i,%i)=",i,j);
            scanf("%f",&a[i][j]);
        }

    Determinante=a[1][1]*a[2][2]-a[2][1]*a[1][2];

    printf("\n\n 1 Calcular el determinante\n");
    printf(" 2 Calcular la inversa\n");
    printf(" 3 Calcular los valores propios\n");
    printf("\nIngrese una opcion (1, 2 %c 3) :",162);
    scanf("%u",&opcion);

    switch(opcion)
    {
        case 1:
            printf("\nEl determinante es %0.1f\n\n",Determinante);
            break;
        case 2:
            if (Determinante!=0.0)
            {
                ainv[1][1]=a[2][2]/Determinante;
```

```

ainv[1][2]=-a[1][2]/Determinante;
ainv[2][1]=-a[2][1]/Determinante;
ainv[2][2]=a[1][1]/Determinante;

for (i=1;i<=n;i++)
for (j=1;j<=n;j++)
printf("\nEl elemento (%i,%i) de la inversa es %0.1f\n",i,j,ainv[i][j]);
}
else printf("\nLa inversa no existe porque el determinante es cero\n\n");
break;
case 3:
Temp=a[1][1]+a[2][2];
Discriminante=pow(Temp,2)-4*Determinante;
if (Discriminante>0)
{
ValPropio1=(Temp+sqrt(Discriminante))/2.;
ValPropio2=(Temp-sqrt(Discriminante))/2.;
printf("\nLos valores propios son %0.1f y %0.1f\n\n",ValPropio1, ValPropio2);
}
else if (Discriminante==0)
{
ValPropio1=Temp/2.;
printf("\nLos valores propios son %0.1f y %0.1f\n\n",ValPropio1, ValPropio1);
}
else
{
ParteReal=Temp/2.;
PartelMagenaria=sqrt(-Discriminante)/2.;
printf("\nLos valores propios son:\n");
printf(" %0.1f + %0.1f i\n", ParteReal, fabs(PartelMagenaria));
printf(" %0.1f - %0.1f i\n\n", ParteReal, fabs(PartelMagenaria));
}
break;
}

system("PAUSE");
return 0;
}

```

Salida 1

```
a(1,1)=-1
a(1,2)=2
a(2,1)=1
a(2,2)=2

1 Calcular el determinante
2 Calcular la inversa
3 Calcular los valores propios
Ingrese una opcion (1, 2 ó 3) :1
El determinante es -4.0
Presione una tecla para continuar . . .
```

Salida 2

```
a(1,1)=1
a(1,2)=2
a(2,1)=3
a(2,2)=4

1 Calcular el determinante
2 Calcular la inversa
3 Calcular los valores propios
Ingrese una opcion (1, 2 ó 3) :2
El elemento (1,1) de la inversa es -2.0
El elemento (1,2) de la inversa es 1.0
El elemento (2,1) de la inversa es 1.5
El elemento (2,2) de la inversa es -0.5
Presione una tecla para continuar . . .
```

Salida 3

```
a<1,1>=3
a<1,2>=2
a<2,1>=-1
a<2,2>=0

1 Calcular el determinante
2 Calcular la inversa
3 Calcular los valores propios

Ingrese una opcion <1, 2 ó 3> :3
Los valores propios son 2.0 y 1.0
Presione una tecla para continuar . . .
```

Observaciones:

1) Se declara la variable `opcion` como `unsigned int opcion`, es decir, un entero sin signo y el especificador de formato usado es `%u`

2) Se calcula la inversa de la matriz A sólo si su determinante es distinto de cero.

67

Escriba un programa en C que muestre por pantalla el polinomio de Lagrange, evaluado en $x = c$, considerando los puntos (x_i, y_i) , $i = 0, 1, 2, \dots, n$, donde el valor de n , los puntos, y el valor de c son ingresados por el usuario. Solicite al usuario el ingreso obligatorio de $c \in (x_0, x_n)$.

(Indicación: El polinomio de Lagrange es

$$L(x) = \sum_{i=0}^n y_i l_i(x)$$

$$\text{con } l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x-x_j}{x_i-x_j}$$

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int i, j, n;
    float c, pol;

    printf("Ingrese el valor de n:");
    scanf("%i",&n);
    float x[n],y[n],l[n];
    for (i=0;i<=n;i++)
    {
        printf("x(%i)=",i);
        scanf("%f",&x[i]);
        printf("y(%i)=",i);
        scanf("%f",&y[i]);
    }
    do
    {
        printf("Ingresa el valor de c:");
        scanf("%f",&c);
    }
    while ((c<=x[0]) || (c>=x[n]));
    for (i=0;i<=n;i++)
    {
        l[i]=1.0;
        for (j=0;j<=n;j++)
        {
            if (i!=j)
                l[i]=l[i]*(c-x[j])/(x[i]-x[j]);
        }
    }
    pol=0.0;
    for (i=0;i<=n;i++)
        pol=pol+y[i]*l[i];
    printf("El polinomio de Lagrange evaluado en x=%0.2f es: %0.2f\n",c,pol);
    printf("\n\n");
    system("PAUSE");
}
```

```
return 0;  
}
```

Salida

```
Ingrese el valor de n:3  
x(0)=1  
y(0)=2.5  
x(1)=2  
y(1)=3.14  
x(2)=3  
y(2)=5.25  
x(3)=4  
y(3)=3.67  
Ingresa el valor de c:2.45  
El polinomio de Lagrange evaluado en x=2.45 es: 4.22  
  
Presione una tecla para continuar . . .
```

Observaciones:

- 1) Dado que $c \in (x_0, x_n)$ debemos obligar al usuario a ingresar un valor en tal intervalo. Esta es la razón de incluir un bucle do .. while con la condición $(c <= x[0]) \ || \ (c >= x[n])$
- 2) La variable pol almacena el valor del polinomio de Lagrange evaluado en $x = c$.

68

Escriba un programa en C que solicite al usuario el ingreso de las estaturas de n alumnos, cuyas alturas no deben superar los 2.25 metros, para luego calcular y mostrar por pantalla la estatura promedio, y si tal promedio es mayor que 1.58, entonces calcule la desviación estándar.
El valor de n es aleatorio y menor o igual que 12.
El proceso anterior completo, debe repetirse cinco veces.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>

int main()
{
    srand(time(NULL));
    int n, i, k;
    for(k=1;k<=5;k++)
    {
        n=rand()%12+1;
        float est[n], suma=0.0, desv=0.0, promedio, desvEst;
        for(i=1;i<=n;i++)
        {
            do
            {
                printf("\nIngrese la estatura %i : ",i);
                scanf("%f",&est[i]);
            }
            while(est[i]>2.25 || est[i]<=0.0);
        }
        for(i=1;i<=n;i++)
            suma=suma+est[i];
        promedio=suma/n;
        printf("\nLa estatura promedio es %0.2f\n",promedio);
        if(promedio>1.58)
        {
            for(i=1;i<=n;i++)
                desv=desv+pow(est[i]-promedio,2);
            desvEst=sqrt(desv/(n-1));
        }
    }
    system("PAUSE");
    return 0;
}
```

Salida

```
Ingrese la estatura 1 : 1.45
Ingrese la estatura 2 : 1.98
La estatura promedio es 1.72
Ingrese la estatura 1 : 2.3
Ingrese la estatura 1 : 2.1
Ingrese la estatura 2 : 1.67
Ingrese la estatura 3 : 1.89
Ingrese la estatura 4 : 1.76
Ingrese la estatura 5 : 1.70
Ingrese la estatura 6 : 2.01
Ingrese la estatura 7 : 2
Ingrese la estatura 8 : 1.57
La estatura promedio es 1.84
Ingrese la estatura 1 : 1.56
Ingrese la estatura 2 : 1.57
La estatura promedio es 1.57
Ingrese la estatura 1 : 2
Ingrese la estatura 2 : 2.1
Ingrese la estatura 3 : 1.78
Ingrese la estatura 4 : 1.98
La estatura promedio es 1.97
Ingrese la estatura 1 : 2.01
Ingrese la estatura 2 : 1.54
Ingrese la estatura 3 : 1.69
Ingrese la estatura 4 : 1.77
Ingrese la estatura 5 : 1.97
Ingrese la estatura 6 : 1.85
Ingrese la estatura 7 : 1.77
Ingrese la estatura 8 : 1.56
La estatura promedio es 1.77

Presione una tecla para continuar . . .
```


Observaciones:

1) Para calcular la desviación estándar es necesario no perder las estaturas ingresadas, por este motivo se utiliza un vector para almacenar tales estaturas.

2) Se utiliza un ciclo for para repetir el proceso 5 veces, pero podríamos haber utilizado un bucle do ... while o uno while ... , con un contador que nos permita saber cuántas veces se ha repetido el proceso.

69

Escriba un programa en C que calcule y muestre por pantalla, con cuatro decimales, el producto interior $\langle A, B \rangle$, con $A \in \mathcal{M}_{m \times n}(\mathbb{R})$ y $B \in \mathcal{M}_{m \times n}(\mathbb{R})$. El usuario es quien ingresa el orden y los elementos de cada matriz. El programa además, debe mostrar por pantalla un mensaje que indique si las matrices son ortogonales o no.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int m, n;
```

```

printf("\nIngrese m :");
scanf("%i",&m);
printf("\nIngrese n :");
scanf("%i",&n);
float A[m][n], B[m][n], Bt[n][m], BtA[n][n];
float traza=0.0;
int i, j, k;
for(i=1;i<=m;i++)
for(j=1;j<=n;j++)
{
printf("\nIngrese A(%i,%i) = ",i,j);
scanf("%f",&A[i][j]);
}
for(i=1;i<=m;i++)
for(j=1;j<=n;j++)
{
printf("\nIngrese B(%i,%i) = ",i,j);
scanf("%f",&B[i][j]);
}
for(i=1;i<=n;i++)
for(j=1;j<=m;j++)
Bt[i][j]=B[j][i];

for(i=1;i<=n;i++)
for(j=1;j<=n;j++)
{
BtA[i][j]=0.0;
for(k=1;k<=m;k++)
BtA[i][j]=BtA[i][j]+Bt[i][k]*A[k][j];
}
for(i=1;i<=n;i++)
traza=traza+BtA[i][i];
printf("\nEl producto interior es %0.4f\n",traza);
if(traza==0.0) printf("\nLas matrices son ortogonales\n");
else printf("\nLas matrices no son ortogonales\n");

system("PAUSE");
return 0;
}

```

Salida

```
Ingrese m :3
Ingrese n :2
Ingrese A<1,1> = 1
Ingrese A<1,2> = -1
Ingrese A<2,1> = 2
Ingrese A<2,2> = 0
Ingrese A<3,1> = 0
Ingrese A<3,2> = 1
Ingrese B<1,1> = -1
Ingrese B<1,2> = -1
Ingrese B<2,1> = 2
Ingrese B<2,2> = 0
Ingrese B<3,1> = 1
Ingrese B<3,2> = 1
El producto interior es 5.0000
Las matrices no son ortogonales
Presione una tecla para continuar . . .
```

Observaciones:

1) $\langle A, B \rangle = \text{traza}(B^T \cdot A)$

2) La variable BtA representa la matriz $B^T \cdot A$, y es de orden n .

70

Usando la orden switch, escriba un programa en C que muestre por pantalla un menú que permita al usuario elegir entre las opciones a) o b) siguientes:

a) Mostrar por pantalla la traza de una matriz de orden m cuyos elementos, en el intervalo con 1 decimal $[-15.2; 14.6]$, son generados aleatoriamente. El valor de m es un número entero mayor que 2 ingresado por el usuario. Debe utilizar una función de nombre traza creada por usted.

b) Muestre por pantalla los primeros 20 polinomios de Laguerre evaluados en 8, y de lo anterior calcule además la suma de los números menores que 10, y finalmente el producto de los números mayores o iguales que 12.

(Ind. : $P_0(x) = 1$; $P_1(x) = 1 - x$; $P_{n+1}(x) = \frac{(2n+1-x)}{n+1} P_n(x) - \frac{n}{n+1} P_{n-1}(x)$)

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

float traza(int, float a[][]);
float traza(int m, float a[m][m])
{
    int i; float sum=0.0;
    for(i=0;i<=m-1;i++)
        sum=sum+a[i][i];
    return sum;
}

float P(float, int);
```

```

float P(float x, int n)
{
    if(n==0) return 1;
    else if(n==1) return 1-x;
    else return (2*n-1.-x)/n*P(x,n-1)-(n-1.)/n*P(x,n-2);
}

```

```

int main()
{
    float x0=8, suma=0.0, prod=1.0;
    int n, m, opcion, i, j;
    float a[100][100];

    printf("\n\t1 Calcular Traza\n\n");
    printf("\n\t2 Polinomios de Laguerre\n\n");
    printf("\n\tIngrese opci%cn : ",162);
    scanf("%i",&opcion);

    switch(opcion)
    {
        case 1:
            do
            {
                printf("\nIngrese m : ");
                scanf("%i",&m);
            }
            while(m<=2);
            srand(time(NULL));
            for(i=0;i<=m-1;i++)
            for(j=0;j<=m-1;j++)
            {
                a[i][j]=(rand()%(146+1+152)-152)/10.0;
            }

            printf("\nLa traza es %0.1f\n",traza(m,a));
            break;

        case 2:
            for(n=0;n<=19;n++)
            {

```

```
printf("\nP(%0.1f,%i)=%0.4f\n",x0,n,P(x0,n));
if(P(x0,n)<10.0) suma=suma+P(x0,n);
if(P(x0,n)>=12.0) prod=prod*P(x0,n);
}
printf("\nLa suma es %0.4f\n",suma);
printf("\nEl producto es %0.4f\n",prod);
break;
}

printf("\n\n");
system("PAUSE");
return 0;
}
```

Salidas 1

```
1 Calcular Traza
2 Polinomios de Laguerre
Ingrese opción : 1
Ingrese m : 2
Ingrese m : 3
La traza es -5.3
Presione una tecla para continuar . . .
```

Salidas 2

```
1 Calcular Traza
2 Polinomios de Laguerre

Ingrese opción : 2
P<8.0,0>=1.0000
P<8.0,1>=-7.0000
P<8.0,2>=17.0000
P<8.0,3>=-12.3333
P<8.0,4>=-9.6667
P<8.0,5>=7.9333
P<8.0,6>=12.0222
P<8.0,7>=1.7873
P<8.0,8>=-8.9556
P<8.0,9>=-10.5443
P<8.0,10>=-3.5387
P<8.0,11>=5.4036
P<8.0,12>=9.9983
P<8.0,13>=8.0868
P<8.0,14>=1.6908
P<8.0,15>=-5.1806
P<8.0,16>=-9.0322
P<8.0,17>=-8.4068
P<8.0,18>=-4.0798
P<8.0,19>=1.7373
La suma es -41.1005
El producto es 204.3778

Presione una tecla para continuar . . .
```

Observaciones:

1) En el prototipo, si tenemos un parámetro que es una matriz, entonces podemos omitir el orden, como se muestra en la línea de código
`float traza(int, float a[][]);`

También es posible omitir el nombre de la matriz, por lo que es lícito escribir
float traza(int, float [][]);

71

Escriba una aplicación de consola, usando matrices y vectores, que calcule y muestre por pantalla, en formato matricial, el producto $A^{-1} \cdot b$, con $A = \begin{cases} 2+i & , i=j \\ j-2 & , i \neq j \end{cases}$ y $b = \begin{bmatrix} 1 \\ -5 \end{bmatrix}$.
Considere que la matriz A es de orden 2, y que el programa debe calcular todo incluida la inversa.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int n=2;
    int A[n][n], b[n], i, j, k, Det;
    float Ainv[n][n], P[n][n];

    b[1]=1; b[2]=-5;

    for(i=1;i<=n;i++)
        for(j=1;j<=n;j++)
            {
                if(i==j) A[i][j]=2+i;
                else A[i][j]=j-2;
            }
}
```



```
Det=A[1][1]*A[2][2]-A[1][2]*A[2][1];
```

```
Ainv[1][1]=(float) A[2][2]/Det;  
Ainv[1][2]=(float) -A[1][2]/Det;  
Ainv[2][1]=(float) -A[2][1]/Det;  
Ainv[2][2]=(float) A[1][1]/Det;
```

```
for(i=1;i<=n;i++)  
for(j=1;j<=n-1;j++)  
{  
P[i][j]=0.0;  
for(k=1;k<=n;k++)  
P[i][j]=P[i][j]+Ainv[i][k]*b[k];  
}
```

```
for(i=1;i<=n;i++)  
{  
printf("\n");  
for(j=1;j<=n-1;j++)  
printf("\t%0.1f",P[i][j]);  
}
```

```
printf("\n\n");  
system("PAUSE");  
return 0;  
}
```

Salida

```
0.3  
-1.2  
Presione una tecla para continuar . . .
```

Observaciones:

1) Si $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$, entonces $A^{-1} = \frac{1}{Det} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}$, donde
 $Det = a_{11}a_{22} - a_{12}a_{21}$

2) La variable Ainv representa la matriz A^{-1} , y la variable P representa el producto $A^{-1} \cdot b$

72

Escriba un programa en C, que muestre por pantalla el resultado de

$$\frac{h}{3} \left[f(x_0) + 4 \sum_{i=1}^{\frac{n}{2}} f(x_{2i-1}) + 2 \sum_{i=1}^{\frac{n-2}{2}} f(x_{2i}) + f(x_n) \right], \text{ con}$$
$$f(x) = x^3 - 1 \text{ y } h = \frac{b-a}{n}$$

El usuario ingresa los valores de a y b , con $a < b$

El n es un número aleatorio par perteneciente al intervalo $[2, 30]$

Además $x_i = a + h i$, para $i = 0, 1, 2, \dots, n$

Note que: $x_0 = a$ y $x_n = b$

Solución:

Código Fuente

```
#include <stdlib.h>
#include <time.h>
#include <math.h>

int main()
{
    srand(time(NULL));
    int n, i;
    float suma1=0.0, suma2=0.0, SumaTotal;
    float a,b, h;
```

```

do
{
n=rand()%29+2;
}
while(n%2!=0);
float x[n];
do
{
printf("\nIngrese a : ");
scanf("%f",&a);
printf("\nIngrese b : ");
scanf("%f",&b);
}
while(a>=b);
h=(b-a)/n;
for(i=0;i<=n-1;i++)
x[i]=a+h*i;
x[n]=b;
if(n==2)
{
SumaTotal=h/3*((pow(x[0],3)-1)+(pow(x[n],3)-1));
}
else
{
for(i=1;i<=n/2;i++)
suma1=suma1+(pow(x[2*i-1],3)-1);
for(i=1;i<=(n-2)/2;i++)
suma2=suma2+(pow(x[2*i],3)-1);
SumaTotal=h/3*((pow(x[0],3)-1)+4*suma1+2*suma2+(pow(x[n],3)-1));
}
printf("\nSuma Total : %0.5f\n\n",SumaTotal);
system("PAUSE");
return 0;
}

```

Salida

```
Ingrese a : 0
Ingrese b : 3
Suma Total : 17.25000
Presione una tecla para continuar . . .
```

Observaciones:

1) Lo que se está calculando en este ejercicio, es una aproximación de la integral definida $\int_a^b (x^3 - 1) dx$, cuyo valor en el programa es SumaTotal

2) El caso $n = 2$, se trabaja por separado para no tener problemas con la suma $\sum_{i=1}^{\frac{n-2}{2}} f(x_{2i})$, pues si $n = 2$, esta suma partiría en $i = 1$ y terminaría en $i = 0$, lo que es imposible.

En el caso $n = 2$, SumaTotal = $\frac{h}{3}[x_0^3 - 1 + x_2^3 - 1]$

3) La variable suma1 almacena la suma $\sum_{i=1}^{\frac{n}{2}} f(x_{2i-1})$, y la variable suma2

almacena la suma $\sum_{i=1}^{\frac{n-2}{2}} f(x_{2i})$

73

Escriba una aplicación en lenguaje C que muestre por pantalla, en formato matricial, la traspuesta de una matriz de orden $m \times n$ cuyos elementos son ingresados por el usuario al igual que el orden.
El programa debe incluir y utilizar una función que trasponga una matriz.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

void trasp(int,int,float a[][],float at[][]);

void trasp(int m, int n,float a[m][n], float at[n][m])
{
    int i,j;
    for(i=1;i<=n;i++)
        for(j=1;j<=m;j++)
            at[i][j]=a[j][i];
}

int main()
{
    int m,n, i, j;

    printf("\nIngrese m : ");
    scanf("%i",&m);
    printf("\nIngrese n : ");
    scanf("%i",&n);

    float a[m][n], at[n][m];

    for(i=1;i<=m;i++)
        for(j=1;j<=n;j++)
        {
            printf("\na(%i,%i) = ",i,j);
            scanf("%f",&a[i][j]);
        }

    trasp(m,n,a,at);

    for(i=1;i<=n;i++)
    {
        printf("\n");
        for(j=1;j<=m;j++)
```

```

    printf("\t%.1f",at[i][j]);
}

printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida

```

Ingrese m : 3
Ingrese n : 2
a<1,1> = 1
a<1,2> = 2
a<2,1> = 3
a<2,2> = 4
a<3,1> = 5
a<3,2> = 6
      1.0    3.0    5.0
      2.0    4.0    6.0
Presione una tecla para continuar . . .

```

Observaciones:

- 1) La variable at es la que almacena la matriz traspuesta.
- 2) C permite parámetros de entrada como de salida. En este caso m , n y a son los parámetros de entrada y at es el parámetro de salida.
- 3) La función trasp no retorna valor, pero sí almacena la traspuesta en el parámetro de salida at. Por este motivo la llamada a la función es trasp(m,n,a,at);
- 4) Cuando un parámetro es una matriz, en la llamada a la función sólo se incluye el nombre de la variable asociada, como se muestra en la línea de código trasp(m,n,a,at);

GUÍA DE PRÁCTICO N° 11

CADENAS DE CARACTERES

Teoría Cadenas de Caracteres

T 43 **Cadenas de Caracteres**

Una cadena de caracteres, o string en inglés, es un arreglo unidimensional de caracteres, por lo que su declaración es similar a la de los vectores. Por ejemplo,

```
char Palabra[20]
```

indica que estamos declarando una variable de nombre Palabra compuesta por 20 caracteres como máximo.

Una cadena de caracteres es una secuencia de caracteres, que pueden ser letras, números u otro símbolo, que termina con un carácter especial llamado fin de cadena o carácter nulo ('\0')

T 44 **Declaración e Inicialización de Cadenas de Caracteres**

Podemos declarar e inicializar una cadena de caracteres del mismo modo que para los vectores.

Por ejemplo,

```
char Mensaje[] = "Este es el primer mensaje";
```

declara la variable Mensaje, de la cual no indicamos el número máximo de caracteres sino que dejamos que C lo determine en forma automática; además la variable Mensaje se inicializa con la cadena de caracteres "Este es el primer mensaje"

Otro ejemplo es

```
char Nombre[5] = {'J','u','a','n','\0'};
```

en donde se especifica cada caracter en forma individual, agregando al final de la cadena el caracter nulo; notemos que es importante reservar un lugar de memoria para el caracter nulo, por tal motivo aparece el número 5 en la declaración y no 4.

Para acceder a la letra J en el ejemplo anterior, se debe escribir Nombre[0], porque los arreglos comienzan con el índice igual a cero. Notemos además que para recorrer los caracteres de la cadena necesitaremos de una estructura iterativa.

T 45 Lectura de una Cadena de Caracteres

Para leer cadenas de caracteres contamos con funciones de la librería stdio.h: scanf, gets y fgets

La función scanf permite leer una palabra, que es una secuencia de caracteres delimitada por espacio, usando el especificador de formato %s

La función gets recibe un único argumento que es la variable relacionada con la cadena de caracteres a leer.

la función fgets permite leer una cadena de caracteres hasta que aparezca un salto de línea (\n) que también es incluido en la cadena. Además se puede indicar el número de caracteres a aceptar en la cadena.

La sintaxis de fgets es

```
fgets(nombre_cadena, numero_caracteres, FILE)
```

donde

nombre_cadena es el nombre que se le ha asignado a la cadena, numero_caracteres es el número de caracteres permitidos para la cadena, lo que hace que fgets sea una función mucho más segura que gets, previniendo un

desbordamiento de búfer, que es un espacio de memoria de almacenamiento temporal.

FILE es la fuente de donde obtendremos la cadena, que puede ser un archivo externo o el teclado. Si es el teclado, entonces FILE se reemplaza por stdin

La función scanf se diferencia de gets y fgets en que scanf sólo captura una cadena de caracteres hasta que aparece el primer espacio. Por ejemplo, si escribimos

```
scanf("%s",Nombre);
```

y el usuario ingresa Marta Miranda, entonces la variable Nombre sólo almacenará Marta. En cambio, si escribimos

```
gets(Nombre);
```

y el usuario ingresa Marta Miranda, entonces la variable Nombre es igual Marta Miranda, incluyendo el apellido.

Ahora si escribimos

```
fgets(Nombre,12,stdin)
```

y el usuario ingresa Marta Miranda por teclado, entonces la variable Nombre almacenará Marta Mirand, pues está limitada a 12 caracteres.

46 Escritura de una Cadena de Caracteres

Para escribir cadenas de caracteres contamos con dos funciones de la librería stdio.h: printf y puts

La función printf permite escribir una palabra, que es una secuencia de caracteres delimitada por espacio, usando el especificador de formato %s

La función puts recibe un único argumento que es la variable relacionada con la cadena de caracteres a escribir por pantalla, seguida de un salto de línea. No es recomendable el uso de la función puts porque no comprueba si nos hemos pasado del espacio de memoria reservado para la variable argumento de la función.

Ejercicios

Cadenas de Caracteres

- [1]** Escriba un programa en C que solicite al usuario el ingreso de su nombre completo y luego lo muestre por pantalla 5 veces en líneas distintas usando colores diferentes para cada línea.
- [2]** Escriba un programa en C que muestre por pantalla las letras de la A a la Z y luego en una línea distinta de la Z a la A, usando ciclos for apropiados.
- [3]** Escriba un programa en C que simule una calculadora simple, es decir, solicite al usuario el ingreso de una operación aritmética (+, -, *, /) y dos números enteros con los cuales calcule y muestre el resultado correspondiente. El programa debe permitirle al usuario continuar con la ejecución hasta que ingrese como operación aritmética una letra N.
- [4]** Use C para contar el número de veces que aparece cada vocal en una oración ingresada por el usuario.
- [5]** Escriba un programa en C que solicite al usuario el ingreso de una clave de 5 caracteres alfanuméricos sin espacios, y emita por pantalla, en color, el mensaje "La palabra clave ingresada es correcta" si la clave coincide con la palabra clave "RLU35" y otro mensaje si no lo es.
- [6]** Escriba un programa en C que solicite al usuario el ingreso de una palabra y determine, mostrando un mensaje en color cian si la palabra ingresada es palíndromo y un mensaje en otro color si no lo es. Asuma que el usuario ingresa la palabra ya sea en minúscula o mayúscula. (Indicación: Una palabra es palíndromo si al leerla de izquierda a derecha o viceversa dice lo mismo, por ejemplo, ANA o ANILINA)

[7] Escriba una aplicación de consola en C que lea una palabra desde teclado y muestre por pantalla la palabra invertida, es decir, si se ingresa la palabra ROMA debe aparecer en pantalla AMOR

[8] Escriba un programa en C que ordene alfabéticamente dos palabras, escritas en minúscula o mayúscula, ingresadas por el usuario.

[9] Escriba un programa en C que cuente el número de palabras en una cadena de caracteres ingresada por el usuario. Una palabra es cualquier cadena de caracteres que no contenga un espacio.

[10] Escriba un programa en C que permita almacenar las respuestas a un examen que tiene 25 preguntas de selección múltiple. Las respuestas pueden ser letras de la A a la E.
Luego el programa debe mostrar por pantalla el número de respuestas A, el número de respuestas B, etc.

[11] Escriba un programa en C que solicite al usuario el ingreso de 5 nombres de ciudades de Chile y muestre por pantalla un mensaje indicando si alguna de las ciudades ingresadas es Santiago.

[12] Escriba un programa en C que solicite al usuario el ingreso de su nombre y si este es GONZALO ACEVEDO, entonces muestre por pantalla el valor

de la función
$$g(t) = \begin{cases} \cos(\sqrt{\pi} \cdot t) & , \text{ si } 0^\circ \leq t \leq 35^\circ \\ 1 - t^5 & , \text{ si } 35^\circ < t < 45^\circ \\ \text{tg}(\sqrt[3]{|t - 15|}) & , \text{ si } 45^\circ \leq t \leq 180^\circ \end{cases}$$

donde t es un ángulo en grados ingresado por GONZALO ACEVEDO.
Si el usuario no es GONZALO ACEVEDO, entonces debe emitir un mensaje por pantalla que muestre solamente el nombre del usuario.

[13] Escriba un programa en C que permita jugar al todo o nada. Un jugador realiza una apuesta indicando la cantidad de dinero y el número del dado al que apuesta. Luego se lanza el dado, si sale el número elegido la persona gana la cantidad apostada y si no pierde lo apostado. En cada lanzamiento se le pregunta al apostador si desea continuar. En caso de no seguir se lleva la cantidad acumulada.

[14] Escriba un programa en C que adivine un número entero, del 1 al 100, pensado por el usuario. A medida que el computador señale las posibles respuestas el usuario debe indicar si la respuesta dada es mayor, menor o igual que el número pensado (en este último caso el computador habrá adivinado el número pensado por el usuario y el programa terminará). Incluya el número total de intentos en la salida por pantalla.

[15] Escriba un programa en C que muestre por pantalla, usando el formato que se indica, la siguiente información:

Estatura (m.)	Peso (kg.)	IMC(kg./m2)	Clasificación	Riesgo
-----	-----	-----	-----	-----
1.50	40	17.8	Rango Normal	Promedio
·	·	·	·	·
·	·	·	·	·

donde el índice de masa corporal (*IMC*) de una persona en función de su estatura (*E*) en metros y de su peso (*P*) en kilogramos, está dado por la fórmula $IMC = \frac{P}{E^2}$

Considere alturas que van desde 1.50 m. hasta 1.80 m. con un incremento de 1 cm., y pesos que van desde 40 kg. hasta 130 kg. con un incremento de 1 kg. Para registrar la clasificación y el riesgo use la tabla siguiente:

Clasificación	IMC(Kg/m ²)	Riesgo
Rango Normal	≤ 24.9	Promedio
Sobrepeso	25 - 29.9	Aumentado
Obesidad grado 1	30 - 34.9	Moderado
Obesidad grado 2	35 - 39.9	Severo
Obesidad grado 3	≥ 40	Muy severo

PROBLEMAS RESUELTOS CADENAS DE CARACTERES

1

Escriba un programa en C que solicite al usuario el ingreso de su nombre completo y luego lo muestre por pantalla 5 veces en líneas distintas usando colores diferentes para cada línea.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <conio2.h>

int main()
{
    char NombreCompleto[50]="";

    printf("\nIngrese su nombre completo :");
    gets(NombreCompleto);

    textbackground(WHITE);
    textcolor(RED);
    printf("\n%s\n",NombreCompleto);

    textcolor(BLUE);
    printf("\n%s\n",NombreCompleto);

    textcolor(CYAN);
    printf("\n%s\n",NombreCompleto);
```

```

textcolor(GREEN);
printf("\n%s\n",NombreCompleto);

textcolor(MAGENTA);
printf("\n%s\n",NombreCompleto);

textcolor(LIGHTGRAY);
printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida

```

Ingrese su nombre completo :Juan Carlos Sandoval
Juan Carlos Sandoval
Juan Carlos Sandoval
Juan Carlos Sandoval
Juan Carlos Sandoval
Juan Carlos Sandoval
Presione una tecla para continuar . . .

```

Observaciones:

- 1) La sentencia `char NombreCompleto[50]="";` declara que trabajaremos con una variable `NombreCompleto` de caracteres (`char`) cuya longitud máxima será de 50 caracteres y que se inicializa con una cadena vacía `""`.
- 2) En este ejercicio la función `gets`, presente en la biblioteca `stdio.h`, permite leer una línea de caracteres desde la consola.
- 3) El formato que se usa para cadenas de caracteres (`string`) es `%s` y el formato que se usa para un caracter es `%c`
- 4) Si hubiésemos usado `scanf` en vez de `gets` el código fuente sería:

```

#include <stdio.h>
#include <stdlib.h>

```

```

#include <conio2.h>

int main()
{
    char NombreCompleto[50]="";

    printf("\nIngrese su nombre completo :");
    scanf("%s",NombreCompleto);

    textbackground(WHITE);
    textcolor(RED);
    printf("\n%s\n",NombreCompleto);

    textcolor(BLUE);
    printf("\n%s\n",NombreCompleto);

    textcolor(CYAN);
    printf("\n%s\n",NombreCompleto);

    textcolor(GREEN);
    printf("\n%s\n",NombreCompleto);

    textcolor(MAGENTA);
    printf("\n%s\n",NombreCompleto);

    textcolor(LIGHTGRAY);
    printf("\n\n");
    system("PAUSE");
    return 0;
}

```

cuya salida es:

```
Ingrese su nombre completo :Juan Carlos Sandoval
Juan
Juan
Juan
Juan
Juan
Presione una tecla para continuar . . .
```

La salida anterior muestra que el nombre completo es cortado al momento de aparecer un espacio, esto muestra que `scanf` procesa mal cadenas de caracteres con espacios y por tal motivo se ha usado `gets`.

Otro detalle importante es que en el `scanf` no se usó el `&` antes de la variable a leer (`NombreCompleto`), porque con las cadenas de caracteres no es necesario incluirlo.

5) Es una buena práctica reservar suficientes lugares para la cadena de caracteres de modo que no excedamos el largo indicado en la declaración.

En el ejercicio, se reservaron 50 lugares de memoria para almacenar el nombre completo del usuario, asumiendo que ningún nombre superará tal cantidad de caracteres.

2

Escriba un programa en C que muestre por pantalla las letras de la A a la Z y luego en una línea distinta de la Z a la A, usando ciclos `for` apropiados.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
```



```

int main()
{
    char letra;

    printf("\n\n");

    for (letra='A';letra<='Z';letra++)
        printf(" %c ",letra);

    printf("\n\n");

    for (letra='Z';letra>='A';letra--)
        printf(" %c ",letra);

    printf("\n\n");
    system("PAUSE");

    return 0;
}

```

Salida

```

A B C D E F G H I J K L M N O P Q R S T U U W X Y Z
Z Y X W U U T S R Q P O N M L K J I H G F E D C B A
Presione una tecla para continuar . . .

```

Observaciones:

- 1) La sentencia `char letra;` declara una variable `letra` que almacenará un caracter.
- 2) Un caracter constante se encierra entre comillas simples, por ejemplo `'A'`, y una cadena de caracteres constante se encierra entre comillas dobles, por ejemplo, `"Hola"`.
- 3) El ejercicio anterior muestra que podemos usar un contador regresivo: `letra=letra-1` o de forma equivalente `letra--`

4) Los caracteres en C pueden ser tratados como números, porque cada caracter tiene efectivamente asociado un número entero, que se rige por un código que habitualmente es el código ASCII; de hecho, el tipo char es un tipo entero.

3

Escriba un programa en C que simule una calculadora simple, es decir, solicite al usuario el ingreso de una operación aritmética (+, -, *, /) y dos números enteros con los cuales calcule y muestre el resultado correspondiente. El programa debe permitirle al usuario continuar con la ejecución hasta que ingrese como operación aritmética una letra N.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main()
{
    char oper[1];
    int num1, num2;
    int suma, resta, mult;
    float division;

    do
    {
        printf("\nIngrese primer n°cmero : ",163);
        scanf("%i",&num1);
        printf("\nIngrese segundo n°cmero : ",163);
```

```

scanf("%i",&num2);

printf("\nIngrese la operaci%cn a realizar (+,-,*,/) : ",162);
scanf("%s",oper);

if(strcmp(oper,"+")==0)
{
    suma=num1+num2;
    printf("\n\t%i + %i = %i\n\n",num1,num2,suma);
}
else if(strcmp(oper,"-")==0)
{
    resta=num1-num2;
    printf("\n\t%i - %i = %i\n\n",num1,num2,resta);
}
else if(strcmp(oper,"*")==0)
{
    mult=num1*num2;
    printf("\n\t%i * %i = %i\n\n",num1,num2,mult);
}
else if(strcmp(oper,"/")==0)
{
    if(num2==0)
        printf("\nDivisi%cn por cero.\n\n",162);
    else
    {
        division= (float) num1/num2;
        printf("\n\t%i / %i = %0.2f\n\n",num1,num2,division);
    }
}
}
while(strcmp(oper,"N")!=0);

printf("\n");
system("PAUSE");
return 0;
}

```

Salida

```
Ingrese primer número : 45
Ingrese segundo número : 33
Ingrese la operación a realizar (<+,-,*,>/> : +
      45 + 33 = 78

Ingrese primer número : 12
Ingrese segundo número : 7
Ingrese la operación a realizar (<+,-,*,>/> : *
      12 * 7 = 84

Ingrese primer número : 3
Ingrese segundo número : 5
Ingrese la operación a realizar (<+,-,*,>/> : /
      3 / 5 = 0.60

Ingrese primer número : 25
Ingrese segundo número : 65
Ingrese la operación a realizar (<+,-,*,>/> : -
      25 - 65 = -40

Ingrese primer número : 2
Ingrese segundo número : 1
Ingrese la operación a realizar (<+,-,*,>/> : N
Presione una tecla para continuar . . .
```

Observaciones:

- 1) La variable oper almacena la operación aritmética a ocupar.
- 2) Producto de la declaración char oper[1]; se hace necesario usar el formato %s en la sentencia scanf("%s",oper);
Lo anterior muestra que C no trata de igual forma a un caracter (char oper) que a una cadena de caracteres de un solo caracter (char oper[1]).

3) En este ejercicio se ocupó la función strcmp, perteneciente a la biblioteca string.h, que permite comparar dos cadenas de caracteres.
La sintaxis es

```
strcmp(cadena1, cadena2)
```

y entrega como resultado un valor igual a cero si ambas cadenas son iguales y un valor distinto de cero si no coinciden.

4

Use C para contar el número de veces que aparece cada vocal en una oración ingresada por el usuario.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main()
{
    char Oracion[80];
    int cuentaA=0, cuentaE=0, cuentaI=0, cuentaO=0, cuentaU=0;
    int i;

    printf("\nIngrese una oración : ",162);
    gets(Oracion);

    for(i=0;i<=strlen(Oracion)-1;i++)
    {
        if(Oracion[i]=='a' || Oracion[i]=='A') cuentaA++;
        if(Oracion[i]=='e' || Oracion[i]=='E') cuentaE++;
        if(Oracion[i]=='i' || Oracion[i]=='I') cuentaI++;
    }
}
```

```

if(Oracion[i]=='o' || Oracion[i]=='O') cuentaO++;
if(Oracion[i]=='u' || Oracion[i]=='U') cuentaU++;
}

printf("\nA = %i",cuentaA);
printf("\nE = %i",cuentaE);
printf("\nI = %i",cuentaI);
printf("\nO = %i",cuentaO);
printf("\nU = %i",cuentaU);

printf("\n\n");
system("PAUSE");
return 0;
}

```

Salida

```

Ingrese una oración : El sol alumbra a todos por igual
A = 4
E = 1
I = 1
O = 4
U = 2
Presione una tecla para continuar . . .

```

Observaciones:

- 1) En este ejercicio asumimos que el usuario ingresa caracteres que estén en el código ASCII básico y no extendido, es decir, por ejemplo ingresa palabras sin tildes.
- 2) Este programa usa la función `strlen`, de la biblioteca `string.h`, que entrega como resultado la longitud de la cadena de caracteres, es decir, el total de caracteres de la cadena que tiene como argumento.
- 3) Podemos acceder a cada caracter de una cadena mediante el uso de un índice, que se mueve desde 0 hasta la longitud de la cadena menos 1. Esto significa que si queremos conocer, por ejemplo, el primer caracter de la cadena `Oracion`, entonces debemos escribir `Oracion[0]` y si quisiéramos escribir el último `Oracion[strlen(Oracion)-1]`

5

Escriba un programa en C que solicite al usuario el ingreso de una clave de 5 caracteres alfanuméricos sin espacios, y emita por pantalla, en color, el mensaje "La palabra clave ingresada es correcta" si la clave coincide con la palabra clave "RLU35" y otro mensaje si no lo es.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <conio2.h>

int main()
{
    char clave[5];

    printf("\nIngrese la clave : ");
    scanf("%s",clave);

    textbackground(WHITE);

    if (strcmp(clave,"RLU35")==0)
    {
        textcolor(GREEN);
        printf("\nLa palabra clave ingresada es correcta\n\n");
    }
    else
    {
        textcolor(LIGHTRED);
        printf("\nLa palabra clave no coincide\n\n");
    }

    system("PAUSE");
    return 0;
}
```

Salida 1

```
Ingrese la clave usando mayúsculas : FER28
La palabra clave no coincide
Presione una tecla para continuar . . .
```

Salida 2

```
Ingrese la clave usando mayúsculas : RLU35
La palabra clave ingresada es correcta
Presione una tecla para continuar . . .
```

Observaciones:

- 1) En este ejercicio asumimos que el usuario ingresa la clave usando mayúsculas.
- 2) Podemos mejorar el programa anterior permitiendo al usuario ingresar la clave usando minúsculas, como muestra el siguiente código fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <conio2.h>

int main()
{
    char clave[5];

    printf("\nIngrese la clave: ");
    scanf("%s",clave);

    textbackground(WHITE);

    if (strcmp(clave,"RLU35")==0 || strcmp(clave,"rlu35")==0)
    {
        textcolor(GREEN);
```



```

printf("\nLa palabra clave ingresada es correcta\n\n");
}
else
{
textcolor(LIGHTRED);
printf("\nLa palabra clave no coincide\n\n");
}

system("PAUSE");
return 0;
}

```

cuyas salidas pueden ser

Salida 1

```

Ingrese la clave: fer27
La palabra clave no coincide
Presione una tecla para continuar . . .

```

Salida 2

```

Ingrese la clave: rlu35
La palabra clave ingresada es correcta
Presione una tecla para continuar . . .

```



6

Escriba un programa en C que solicite al usuario el ingreso de una palabra y determine, mostrando un mensaje en color cian si la palabra ingresada es palíndromo y un mensaje en otro color si no lo es. Asuma que el usuario ingresa la palabra ya sea en minúscula o mayúscula. (Indicación: Una palabra es palíndromo si al leerla de izquierda a derecha o viceversa dice lo mismo, por ejemplo, ANA o ANILINA)

Solución:

Código Fuente

```
##include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <conio2.h>

int main()
{
    char Palabra[15];
    int Palindromo=1, Largo, i;

    printf("\nIngrese una palabra : ");
    scanf("%s",Palabra);

    Largo=strlen(Palabra);

    for(i=0;i<=Largo-1;i++)
        if(Palabra[i]!=Palabra[Largo-i-1]) Palindromo=0;

    textbackground(WHITE);
    if(Palindromo==1)
    {
        textcolor(CYAN);
        printf("\n%s es pal%ndromo\n",Palabra,161);
    }
    else
    {
        textcolor(LIGHTRED);
        printf("\n%s no es pal%ndromo\n",Palabra,161);
    }

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida 1

```
Ingrese una palabra : anilina
anilina es palíndromo
Presione una tecla para continuar . . .
```

Salida 2

```
Ingrese una palabra : entregar
entregar no es palíndromo
Presione una tecla para continuar . . .
```

Observaciones:

- 1) En este ejercicio asumimos que el usuario ingresa una palabra con a lo más 15 letras, tal como lo indica la declaración `char Palabra[15];`
- 2) Para resolver este ejercicio hacemos uso de una variable `Palindromo` que sirve como banderín (flag) indicador de si la palabra es palíndromo (valor 1) o no (valor 0).
- 3) Una palabra será palíndromo cuando el primer caracter coincide con el último, el segundo con el penúltimo, y así sucesivamente; esto es justamente lo que se hace en el ciclo, pero detectando si algún par de letras correspondientes no son iguales, en cuyo caso la palabra no será palíndromo.

```
for(i=0;i<=Largo-1;i++)
    if(Palabra[i]!=Palabra[Largo-i-1]) Palindromo=0;
```

Si al salir de este ciclo todos los caracteres correspondientes coinciden entonces la variable `Palindromo` seguirá manteniendo el valor 1 dado en la declaración al inicio del programa, y esto significa que la palabra es finalmente palíndromo.

8

Escriba un programa en C que ordene alfabéticamente dos palabras, escritas en minúscula o mayúscula, ingresadas por el usuario.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main()
{
    char Palabra1[15], Palabra2[15];

    printf("\nIngrese una palabra : ");
    scanf("%s",Palabra1);

    printf("\nIngrese otra palabra : ");
    scanf("%s",Palabra2);

    if(strcmp(Palabra1,Palabra2)>0) printf("\n%s\t%s\n",Palabra2,Palabra1);
    else printf("\n%s\t%s\n",Palabra1,Palabra2);

    printf("\n\n");
    system("PAUSE");
    return 0;
}
```

Salida 1

```
Ingrese una palabra : permitir
Ingrese otra palabra : permitido
permitido      permitir
Presione una tecla para continuar . . .
```

Salida 2

```
Ingrese una palabra : atrofiado
Ingrese otra palabra : perezoso
atrofiado      perezoso
Presione una tecla para continuar . . .
```

Observación:

La función strcmp entrega un resultado positivo cuando Palabra1 es "mayor" que Palabra 2, esto significa desde el punto de vista alfabético que Palabra2 debe ir primero y luego Palabra1, porque por ejemplo la letra b tiene un valor mayor que la letra a.

9

Escriba un programa en C que cuente el número de palabras en una cadena de caracteres ingresada por el usuario. Una palabra es cualquier cadena de caracteres que no contenga un espacio.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
```

```

#include <string.h>

int main()
{
    char oracion[60];
    int i, palabra=1;

    printf("\nIngrese una oraci%cn : ",162);
    gets(oracion);

    for(i=0;i<=strlen(oracion)-1;i++)
        if(oracion[i]==' ') palabra++;

    printf("\nEl total de palabras en la oraci%cn ingresada es %i\n",162,palabra);
    system("PAUSE");
    return 0;
}

```

Salida 1

```

Ingrese una oración : Este es un espacio para escribir lo que quiera
El total de palabras en la oración ingresada es 9
Presione una tecla para continuar . . .

```

Salida 2

```

Ingrese una oración : Ahora, un ejemplo demostrativo
El total de palabras en la oración ingresada es 4
Presione una tecla para continuar . . .

```

Observación:

El contador palabra cuenta el total de espacios en la oración, por tal motivo se inicializa en 1, porque después del último espacio aparecerá una palabra que también debemos contar.

Escriba un programa en C que permita almacenar las respuestas a un examen que tiene 25 preguntas de selección múltiple. Las respuestas pueden ser letras de la A a la E.
Luego el programa debe mostrar por pantalla el número de respuestas A, el número de respuestas B, etc.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    const n=25;
    char x[n];
    int i;
    int cuentaA=0, cuentaB=0, cuentaC=0, cuentaD=0, cuentaE=0;
    int cuentaOtras=0;

    for (i=0;i<=n-1;i++)
    {
        printf("Ingrese respuesta pregunta %i : ",i+1);
        scanf("%s",&x[i]);
    }
    for (i=0;i<=n-1;i++)
    switch(x[i])
    {
        case 'A': cuentaA=cuentaA+1; break;
        case 'B': cuentaB=cuentaB+1; break;
        case 'C': cuentaC=cuentaC+1; break;
        case 'D': cuentaD=cuentaD+1; break;
        case 'E': cuentaE=cuentaE+1; break;
        default: cuentaOtras++; break;
    }
    printf("\nTotal de respuestas A : %i",cuentaA);
    printf("\nTotal de respuestas B : %i",cuentaB);
```

```

printf("\nTotal de respuestas C : %i",cuentaC);
printf("\nTotal de respuestas D : %i",cuentaD);
printf("\nTotal de respuestas E : %i",cuentaE);
printf("\nTotal de otras respuestas : %i\n\n",cuentaOtras);

system("PAUSE");
return 0;
}

```

Salida

```

Ingrese respuesta pregunta 1 : A
Ingrese respuesta pregunta 2 : D
Ingrese respuesta pregunta 3 : A
Ingrese respuesta pregunta 4 : E
Ingrese respuesta pregunta 5 : C
Ingrese respuesta pregunta 6 : A
Ingrese respuesta pregunta 7 : B
Ingrese respuesta pregunta 8 : C
Ingrese respuesta pregunta 9 : B
Ingrese respuesta pregunta 10 : B
Ingrese respuesta pregunta 11 : C
Ingrese respuesta pregunta 12 : C
Ingrese respuesta pregunta 13 : A
Ingrese respuesta pregunta 14 : E
Ingrese respuesta pregunta 15 : D
Ingrese respuesta pregunta 16 : D
Ingrese respuesta pregunta 17 : A
Ingrese respuesta pregunta 18 : B
Ingrese respuesta pregunta 19 : B
Ingrese respuesta pregunta 20 : B
Ingrese respuesta pregunta 21 : C
Ingrese respuesta pregunta 22 : D
Ingrese respuesta pregunta 23 : D
Ingrese respuesta pregunta 24 : E
Ingrese respuesta pregunta 25 : E

Total de respuestas A : 5
Total de respuestas B : 6
Total de respuestas C : 5
Total de respuestas D : 5
Total de respuestas E : 4
Total de otras respuestas : 0

Presione una tecla para continuar . . .

```

Observaciones:

1) En este ejercicio asumimos que el usuario ingresa caracteres que estén en mayúscula. Si el usuario ingresa las respuestas en minúscula estas no se cuentan como aparece en la siguiente salida:


```

Ingrese respuesta pregunta 1 : a
Ingrese respuesta pregunta 2 : a
Ingrese respuesta pregunta 3 : a
Ingrese respuesta pregunta 4 : b
Ingrese respuesta pregunta 5 : e
Ingrese respuesta pregunta 6 : e
Ingrese respuesta pregunta 7 : d
Ingrese respuesta pregunta 8 : d
Ingrese respuesta pregunta 9 : c
Ingrese respuesta pregunta 10 : c
Ingrese respuesta pregunta 11 : a
Ingrese respuesta pregunta 12 : e
Ingrese respuesta pregunta 13 : d
Ingrese respuesta pregunta 14 : c
Ingrese respuesta pregunta 15 : e
Ingrese respuesta pregunta 16 : e
Ingrese respuesta pregunta 17 : d
Ingrese respuesta pregunta 18 : d
Ingrese respuesta pregunta 19 : c
Ingrese respuesta pregunta 20 : b
Ingrese respuesta pregunta 21 : b
Ingrese respuesta pregunta 22 : b
Ingrese respuesta pregunta 23 : a
Ingrese respuesta pregunta 24 : d
Ingrese respuesta pregunta 25 : d

Total de respuestas A : 0
Total de respuestas B : 0
Total de respuestas C : 0
Total de respuestas D : 0
Total de respuestas E : 0
Total de otras respuestas : 25

Presione una tecla para continuar . . .

```

Para corregir lo anterior se puede usar la función `toupper(car)` que convierte a mayúscula el carácter `car`. Esta función pertenece a la biblioteca `ctype.h`. El nuevo código fuente es:

```

#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>

int main()
{
    const n=25;
    char x[n];
    int i;
    int cuentaA=0, cuentaB=0, cuentaC=0, cuentaD=0, cuentaE=0;
    int cuentaOtras=0;

    for (i=0;i<=n-1;i++)
    {

```

```

printf("Ingrese respuesta pregunta %i : ",i+1);
scanf("%s",&x[i]);
}
for (i=0;i<=n-1;i++)
switch(toupper(x[i]))
{
case 'A': cuentaA=cuentaA+1; break;
case 'B': cuentaB=cuentaB+1; break;
case 'C': cuentaC=cuentaC+1; break;
case 'D': cuentaD=cuentaD+1; break;
case 'E': cuentaE=cuentaE+1; break;
default: cuentaOtras++; break;
}
printf("\nTotal de respuestas A : %i",cuentaA);
printf("\nTotal de respuestas B : %i",cuentaB);
printf("\nTotal de respuestas C : %i",cuentaC);
printf("\nTotal de respuestas D : %i",cuentaD);
printf("\nTotal de respuestas E : %i",cuentaE);
printf("\nTotal de otras respuestas : %i\n\n",cuentaOtras);

system("PAUSE");
return 0;
}

```

2) La función que convierte a minúscula el caracter car es tolower(car) también presente en la biblioteca ctype.h

Las funciones mencionadas solo convierten caracteres y no cadenas de caracteres; para lograr esto último es necesario convertir cada caracter de la cadena.

Escriba un programa en C que solicite al usuario el ingreso de su nombre y si este es GONZALO ACEVEDO, entonces muestre por pantalla el valor de la

$$\text{función } g(t) = \begin{cases} \cos(\sqrt{\pi} \cdot t) & , \text{ si } 0^\circ \leq t \leq 35^\circ \\ 1 - t^5 & , \text{ si } 35^\circ < t < 45^\circ \\ \text{tg}(\sqrt[3]{|t - 15|}) & , \text{ si } 45^\circ \leq t \leq 180^\circ \end{cases}$$

donde t es un ángulo en grados ingresado por GONZALO ACEVEDO.

Si el usuario no es GONZALO ACEVEDO, entonces debe emitir un mensaje por pantalla que muestre solamente el nombre del usuario.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>

int main()
{
    char nombre[50]="";
    float t, rad;
    printf("\nIngrese su nombre : ");
    gets(nombre);
    if (strcmp(nombre,"GONZALO ACEVEDO")==0)
    {
        printf("\nIngrese t :");
        scanf("%f",&t);
        rad=t*M_PI/180;
        if (t>=0 && t<=35)
        {
            printf("\nEl valor para t = %0.0f es %0.2f\n",t,cos(sqrt(M_PI)*rad));
        }
        else if ((t>35) && (t<45))
        {
            printf("\nEl valor para t = %0.0f es %0.2f\n",t,1-pow(t,5));
        }
    }
}
```

```

    }
    else if(t>=45 && t<=180)
        printf("\nEl valor para t = %0.0f es %0.2f\n",t, tan(pow(fabs(rad-15),1/3.)));
    }
    else
        printf("\nSu nombre es %s\n", nombre);
    system("PAUSE");
    return 0;
}

```

Salida 1

```

Ingrese su nombre : GONZALO ACEUEDO
Ingrese t :3
El valor para t = 3 es 1.00
Presione una tecla para continuar . . .

```

Salida 2

```

Ingrese su nombre : FERNANDO PARRA
Su nombre es FERNANDO PARRA
Presione una tecla para continuar . . .

```

Observación:

En este ejercicio asumimos que el usuario ingresa caracteres que estén en mayúscula.

GUÍA DE PRÁCTICO N° 12

ARCHIVOS

Teoría Archivos

47 Manejo de Archivos o Ficheros

Al igual que existen funciones para mostrar información por pantalla y recibir información por teclado, también existen funciones que permiten leer y escribir en archivos. Trabajaremos sólo con archivos de texto, que son documentos que el usuario puede leer con un programa como por ejemplo el bloc de notas de Windows.

Los pasos a seguir para trabajar con archivos de texto externos son los siguientes:

- 1) Declarar una variable de tipo **FILE ***, que contendrá un puntero a un nombre de archivo interno
- 2) Abrir el archivo declarado en el punto 1), usando la función **fopen**
- 3) Sobre el archivo, realizar operaciones de lectura con la función **fscanf**, y escritura con la función **fprintf**
- 4) Cerrar el archivo usando la función **fclose**

48 Apertura de un Archivo

Para abrir un archivo es necesario el uso de la función `fopen` incluida en la librería `stdio.h`

La función `fopen` contiene dos argumentos:

a) el primero indica la ruta completa relacionada con el archivo externo que manejaremos. Tal ruta debe ir entre comillas dobles.

b) el segundo argumento describe el uso que se le dará al archivo; existen tres parámetros básicos:

"r" archivo de lectura (read); el archivo debe existir

"w" archivo de escritura (write); el archivo se crea si no existe y se sobrescribe si existe

"a" archivo de escritura al final del contenido (append); se crea si no existe

Además el resultado de la función `fopen` es asignado a la variable puntero que contiene el nombre interno a usar en nuestro código fuente, declarado con `FILE *`

Por ejemplo,

```
FILE * MiArchivoDeLectura;  
MiArchivoDeLectura=fopen("c:/temp/hola.txt","r");
```

son sentencias que permiten enlazar el nombre interno `MiArchivoDeLectura` con el archivo externo de lectura `hola.txt`, que ya existe en el disco `c:` del computador y en el directorio `temp` de este.

49 **Lectura y Escritura de Archivos**

Para leer datos de un archivo existente en algún medio de almacenamiento, podemos usar la función `fscanf` cuyo uso es exactamente igual al de `scanf`, con la diferencia que incluye un argumento adicional que es el nombre interno, y además la información no es ingresada por teclado, sino que es leída desde el archivo externo ya existente.

Para escribir datos a un archivo, podemos usar la función `fprintf` cuyo uso es exactamente igual al de `printf`, con la diferencia que incluye un argumento adicional que es el nombre interno, y además la información no aparece en pantalla sino que es guardada en el archivo externo considerado.

En un programa podemos realizar operaciones solo de lectura, solo de escritura o ambas. Si en una aplicación trabajamos con archivos de lectura y escritura, entonces deberemos declarar dos variables de tipo `FILE *`, una para lectura y otra para escritura; además deberemos incluir en correspondencia dos sentencias `fopen` y dos sentencias `fclose`.

Las funciones `fscanf` y `fprintf` están incluidas en la biblioteca `stdio.h`

Usando el ejemplo mencionado en el ítem teórico anterior, se tiene que

```
fscanf(MiArchivoDeLectura,"%f",&nota);
```

permite leer el primer dato presente en el archivo `hola.txt` el que es almacenado en la variable de tipo real `nota`

50 **Cierre de Archivos**

Los archivos de lectura y de escritura deben ser cerrados una vez que ya han sido utilizados, para ello se hace uso de la función `fclose` incluida en la biblioteca `stdio.h`. La función `fclose` contiene solo un argumento que es el nombre interno, declarado con `FILE *`, asociado al archivo externo.

Siguiendo con el ejemplo mencionado anteriormente, tenemos que

```
fclose(MiArchivoDeLectura);
```

cierra el archivo externo `hola.txt`, y desde este punto ya no se podrá leer desde él, a menos que lo abramos de nuevo.

Ejercicios Archivos

[1] Dada una matriz cuadrada de orden n , calcule y muestre por pantalla, con 5 decimales, la matriz que resulta de restar a cada elemento de la matriz la media aritmética de su fila.

El valor de n y los elementos de la matriz deben ser leídos desde un archivo externo MATRIZ.DAT. El archivo MATRIZ.DAT debe ser generado por otro programa en C.

Además, debe guardar en un archivo externo de nombre SALIDA.SAL el valor de n y la matriz original.

[2] Se sabe que existe una relación funcional del tipo $y = x^3 + 2x - 3$ entre una variable independiente x , y una variable dependiente y .

Prepare un programa que dé la opción de ver en pantalla o guardar en un archivo externo DATOS.DAT, la ecuación y los valores de x y y , en un rango e incremento ingresados por el usuario.

[3] Escriba un programa en C que muestre por pantalla, usando formato matricial, la inversa de una matriz triangular superior de orden 10 leída desde un archivo externo de nombre MATRIZ.TXT

[4] Escriba un programa en C que lea desde un archivo MATRIZ.DOC el orden y los elementos de una matriz y muestre por pantalla la norma euclidiana indicando si la matriz es unitaria o no.

Un ejemplo de archivo MATRIZ.DOC es:

```
1
3
-1
2
0
```

donde el primer número indica el número de filas, el segundo número indica el número de columnas y los restantes números son los elementos de la matriz.

[5] Escriba un programa en C que lea desde un archivo de texto externo de nombre NUMEROS.TXT tres números reales y muestre por pantalla el promedio de ellos con un decimal.

[6] Escriba un programa en C que lea desde el archivo MATRIZ.DOC los elementos de una matriz cuadrada de orden 2 y luego guarde en un archivo TRAZA.TXT los elementos de la matriz junto con su traza.

[7] Para calcular la integral definida $I = \int_a^b f(x) dx$ se procede a dividir el intervalo $[a, b]$ en un número n par de partes iguales de longitud h , obteniendo $(n + 1)$ puntos x_i , con $i = 0, 1, 2, \dots, n$ y donde $x_0 = a$ y $x_n = b$.

Considerando $h = \frac{b-a}{n}$ y $f(x_i) = y_i$, se tiene $I = \frac{h}{3} [y_0 + 4IM + 2PA + y_n]$

donde $IM = y_1 + y_3 + y_5 + \dots + y_{n-1}$; $PA = y_2 + y_4 + y_6 + \dots + y_{n-2}$

Prepare un programa que permita:

a) Calcular la integral $\int_a^b e^{x^2} \text{sen}(x) dx$ para 5 intervalos $[a, b]$ distintos ingresados por el usuario, usando el método anterior ("Método de Simpson"). Utilice $n = 10$.

b) Generar un archivo de texto con la estructura siguiente, usando los resultados de la parte a)

- Método de Simpson -

Intervalo	Valor integral
[xx , xx]	xxxxxxxxxxxxx
:	:
:	:
Número de subdivisiones: xx	
Función : xxxxxxxx	

[8] Escriba un programa en C que guarde en un archivo externo de nombre PARES.SAL los números pares positivos menores que 1000. Los números deben aparecer uno al lado del otro separados por una tabulación.

[9] a) ¿Qué son los archivos de texto externos a C? ¿Para qué sirven? ¿Cómo se declaran?

b) ¿De cuántos modos se puede abrir un archivo externo?

c) Los archivos de lectura, ¿cómo se crean? Los archivos de escritura, ¿cómo se crean?

[10] Prepare un programa en C que haciendo uso de color y un decimal para los resultados

a) Calcule y muestre por pantalla, usando formato matricial, la matriz

$$\begin{bmatrix} \langle \varphi_1, \varphi_1 \rangle & \langle \varphi_1, \varphi_2 \rangle \\ \langle \varphi_2, \varphi_1 \rangle & \langle \varphi_2, \varphi_2 \rangle \end{bmatrix}$$

donde $\varphi_1(x) = 1$, $\varphi_2(x) = x$ y $\langle f, g \rangle = \sum_{i=1}^n f(x_i) g(x_i)$

El valor de n y el de los x_i , $i = 1, 2, \dots, n$ deben ser leídos desde un archivo externo **DATOS.DAT**. Utilice funciones en forma apropiada.

b) Calcule y guarde en un archivo externo **DETER.SAL** el determinante de la matriz calculada en la parte a)

[11] Supongamos que es posible escribir una matriz A cuadrada de orden n como el producto $A = LU$, donde L es una matriz triangular inferior con unos en la diagonal principal, y U es una matriz triangular superior.

La forma general para la obtención de los coeficientes de las matrices L y U está dada por:

$$u_{1j} = a_{1j}, j = 1, 2, \dots, n$$

$$l_{i1} = \frac{a_{i1}}{u_{11}}, i = 2, \dots, n$$

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj}, i = 2, 3, \dots, n; j = i, i+1, \dots, n$$

$$l_{ij} = \frac{a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj}}{u_{jj}}, j = 2, 3, \dots, n-1; i = j+1, j+2, \dots, n$$

Escriba un programa en C de nombre LU.C que lea desde pantalla una matriz cuadrada de orden n , donde n es ingresado por el usuario, y guarde en un archivo externo de nombre LU.TXT la matriz A y las matrices L y U , considerando 2 decimales.

[12] Escriba un programa en C, que guarde en un archivo externo de nombre TABLAS.TXT las tablas del 1 al 12 considerando el siguiente formato de salida :

TABLA DEL 1

1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5
1 x 6 = 6
1 x 7 = 7
1 x 8 = 8
1 x 9 = 9
1 x 10 = 10
1 x 11 = 11
1 x 12 = 12

TABLA DEL 2

2 x 1 = 2
2 x 2 = 4
:
:

[13] Escriba un programa en C que guarde en un archivo de texto y muestre por pantalla, el polinomio de Lagrange, evaluado en $x = c$, considerando los puntos (x_i, y_i) , $i = 0, 1, 2, \dots, n$, donde el valor de n , los puntos, y el valor de c son ingresados por el usuario. Solicite al usuario el ingreso obligatorio de $c \in (x_0, x_n)$.

(Indicación: El polinomio de Lagrange es

$$L(x) = \sum_{i=0}^n y_i l_i(x)$$

$$\text{con } l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x-x_j}{x_i-x_j}$$

[14] Escriba un programa en C que muestre por pantalla, un mensaje indicando si la distancia entre las matrices A y B es mayor que π .

Suponga que $A, B \in \mathcal{M}_3(\mathbb{R})$ son leídas desde un archivo externo MATRIZ.TXT.

Defina y use una función que calcule el máximo entre dos números reales.

Indicación: $d(A, B) = \|B - A\|$, donde

$$\|P\| = \text{maximo} \left\{ \sum_{j=1}^3 |p_{1j}|, \sum_{j=1}^3 |p_{2j}|, \sum_{j=1}^3 |p_{3j}| \right\}$$

[15] Escriba un programa en C que muestre por pantalla un menú, usando la orden SWITCH, que permita al usuario elegir entre las opciones a) ó b) siguientes:

a) Mostrar por pantalla si una matriz de orden n es simétrica o no, cuyos elementos en el intervalo $[-5.1; 14.8]$, son generados aleatoriamente. El valor de n es leído desde `c:/orden.dat`. Use una función apropiada creada por usted.

b) Guardar en `c:/fib.txt` los primeros 20 números de la sucesión de Fibonacci, además la suma de los pares, y finalmente el producto de los impares mayores que 1 y menores o iguales que 17.

[16] Escriba un programa en C que lea desde un archivo externo `CURSO.DAT` las notas de la primera prueba de 15 alumnos de un curso y muestre por pantalla, con 1 decimal, el promedio y la desviación estándar de las notas del curso.

[17] Escriba un programa en C que genere aleatoriamente 500 números enteros positivos que estén en el intervalo $[a, b]$, donde a y b son números enteros positivos ingresados por el usuario. El programa debe verificar que a sea menor que b , y que b no sea superior a 50.

El programa debe también depositar estos números en un arreglo, cuyo primer índice es 0, y mostrar por pantalla la desviación estándar de los elementos con índice impar. Además debe guardar en el directorio raíz del disco, en un archivo `PROD.DOC`, usando el formato que se indica, la siguiente información:

$x[0] = xx$	$x[2] = xx$	$x[4] = xx$	$x[6] = xx$
$x[8] = xx$	$x[10] = xx$	$x[12] = xx$	$x[14] = xx$
$x[16] = xx$	$x[18] = xx$	$x[20] = xx$	$x[22] = xx$
\vdots	\vdots	\vdots	\vdots
\vdots	\vdots	\vdots	\vdots

La suma de los elementos de índice par es : xx

OBS.: xx representa los valores generados por el programa.

$$\text{Promedio} = \bar{x} = \frac{\sum_{i=1}^n x_i}{n} \quad \text{Desviación estándar} = s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}}$$

[18] Escriba un programa en C que muestre por pantalla, en color, las soluciones de la ecuación cúbica $dx^3 + ax^2 + bx + c = 0$, donde los coeficientes d, a, b y c son ingresados por el usuario, usando el siguiente esquema de trabajo:

Para la ecuación cúbica $x^3 + ax^2 + bx + c = 0$ (si el coeficiente que acompaña a x^3 no es 1, entonces divida la ecuación por tal coeficiente), con $a, b, c \in \mathbb{R}$, primero calcule

$$Q \equiv \frac{a^2 - 3b}{9} \quad R \equiv \frac{2a^3 - 9ab + 27c}{54}$$

Si $R^2 < Q^3$, entonces la ecuación posee tres raíces reales, las cuales son:

$$\begin{aligned} x_1 &= -2\sqrt{Q} \cos\left(\frac{\theta}{3}\right) - \frac{a}{3} \\ x_2 &= -2\sqrt{Q} \cos\left(\frac{\theta + 2\pi}{3}\right) - \frac{a}{3} \\ x_3 &= -2\sqrt{Q} \cos\left(\frac{\theta - 2\pi}{3}\right) - \frac{a}{3} \end{aligned}$$

donde $\theta = \text{Arccos}\left(\frac{R}{\sqrt{Q^3}}\right)$.

Si $R^2 \geq Q^3$, entonces las tres raíces son:

$$\begin{aligned} x_1 &= (A + B) - \frac{a}{3} \\ x_2 &= -\frac{1}{2}(A + B) - \frac{a}{3} + i\frac{\sqrt{3}}{2}(A - B) \\ x_3 &= -\frac{1}{2}(A + B) - \frac{a}{3} - i\frac{\sqrt{3}}{2}(A - B) \end{aligned}$$

donde $A = -\text{signo}(R) \left[|R| + \sqrt{R^2 - Q^3} \right]^{1/3}$

$$B = \begin{cases} Q/A, & \text{si } A \neq 0 \\ 0, & \text{si } A = 0 \end{cases}$$

$$\text{signo}(R) = \begin{cases} +1, & \text{si } R \geq 0 \\ -1, & \text{si } R < 0 \end{cases}$$

Además, guarde en un archivo externo RAICES.DOC los coeficientes y las soluciones de la ecuación cúbica. Defina y use una función Signo. (En C, Arccos es acos, presente en math.h)

[19] Suponga que : $f(x) = \cos(x^3) + 2\text{sen}(x) - 3\sqrt[5]{x^2}$

Escriba un programa en C que dé la opción al usuario de ver en pantalla o guardar en un archivo externo DATOS.DAT, los valores de x y $f(x)$, para $x = x_0, x_0 + h, x_0 + 2h, \dots, x_0 + nh$, con x_0, h y n leídos desde un archivo de

texto RANGO.TXT presente en el directorio raíz del disco. Suponga que x está expresado en grados.

Es obligatorio el uso de, al menos, una función.

Indique claramente el formato del archivo RANGO.TXT.

[20] Escriba un programa en C que permita al usuario, usando el comando SWITCH, elegir entre una de las siguientes opciones:

a) calcular y mostrar por pantalla, en color verde, el área y el perímetro de un rectángulo cuyas longitudes de los lados son ingresados por el usuario

b) calcular y guardar en un archivo externo AREA.TXT el área de un círculo cuyo radio es un número aleatorio entero entre 2 y 27 inclusive

c) mostrar por pantalla, en color azul, un mensaje indicando si un número entero ingresado por el usuario es múltiplo de 3 y 5.

[21] Escriba un programa en C que calcule, muestre por pantalla y guarde en un archivo de texto, en formato matricial, la inversa de una matriz no singular de orden 2 ingresada por el usuario. Para calcular el determinante debe usar una función de nombre Det.

[22] Usando la orden switch, escriba un programa en C que muestre por pantalla un menú que permita al usuario elegir una de las opciones 1) ó 2) siguientes:

1) Guardar en un archivo externo, la traza de una matriz de orden k cuyos elementos, en el intervalo $[-15.2; 14.6]$, son ingresados por el usuario. El valor de k es un número entero mayor que 2, ingresado por el usuario. Debe implementar una función traza.

2) Muestre por pantalla la suma de las primeras m componentes de un vector de n componentes, con $m \leq n$. El usuario es quien ingresa el valor de m y las componentes del vector, y el valor de n es un número que se obtiene desde el archivo num.txt.

PROBLEMAS RESUELTOS ARCHIVOS

13

Escriba un programa en C que guarde en un archivo de texto y muestre por pantalla el polinomio de Lagrange, evaluado en $x = c$, considerando los puntos (x_i, y_i) , $i = 0, 1, 2, \dots, n$, donde el valor de n , los puntos, y el valor de c son ingresados por el usuario. Solicite al usuario el ingreso obligatorio de $c \in (x_0, x_n)$.

(Indicación: El polinomio de Lagrange es

$$L(x) = \sum_{i=0}^n y_i l_i(x)$$

$$\text{con } l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x-x_j}{x_i-x_j}$$

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int i, j, n;
    float c, pol;
    FILE *Lag;
    Lag=fopen("c:/temp/lagrange.txt","w");
    printf("Ingrese el valor de n:");
```

```

scanf("%i",&n);
float x[n],y[n],l[n];
for (i=0;i<=n;i++)
{
    printf("x(%i)=",i);
    scanf("%f",&x[i]);
    printf("y(%i)=",i);
    scanf("%f",&y[i]);
}
do
{
    printf("Ingresa el valor de c:");
    scanf("%f",&c);
}
while ((c<=x[0]) || (c>=x[n]));
for (i=0;i<=n;i++)
{
    l[i]=1.0;
    for (j=0;j<=n;j++)
    {
        if (i!=j)
            l[i]=l[i]*(c-x[j])/(x[i]-x[j]);
    }
}
pol=0.0;
for (i=0;i<=n;i++)
    pol=pol+y[i]*l[i];
printf("El polinomio de Lagrange evaluado en x=%0.2f es: %0.2f\n",c,pol);
fprintf(Lag,"El polinomio de Lagrange evaluado en x=%0.2f es: %0.2f\n",c,pol);
fclose(Lag);
system("PAUSE");
return 0;
}

```


Salida

```
Ingrese el valor de n:7
x(0)=0
y(0)=0.567
x(1)=1
y(1)=2.356
x(2)=2
y(2)=3.678
x(3)=3
y(3)=-1.276
x(4)=4
y(4)=-2.678
x(5)=5
y(5)=-3.362
x(6)=6
y(6)=0.211
x(7)=7
y(7)=0.569
Ingresa el valor de c:3.77
El polinomio de Lagrange evaluado en x=3.77 es: -2.56
Presione una tecla para continuar . . .
```

Archivo lagrange.txt

```
El polinomio de Lagrange evaluado en x=3.77 es: -2.56
```

Observaciones:

- 1) El valor de n indica el número de puntos más uno.
- 2) El valor de c debe ser un número en el intervalo $[0, 7]$ para la salida anterior, por tal motivo se ingresa el valor 3.77

3) El vector $l[i]$ se corresponde con $l_i(x) = \prod_{j=0, j \neq i}^n \frac{x-x_j}{x_i-x_j}$

4) La variable `pol` almacena el valor de $L(x) = \sum_{i=0}^n y_i l_i(x)$

20

Escriba un programa en C que permita al usuario, usando el comando SWITCH, elegir entre una de las siguientes opciones:

- a) calcular y mostrar por pantalla, en color verde, el área y el perímetro de un rectángulo cuyas longitudes de los lados son ingresados por el usuario
- b) calcular y guardar en un archivo externo AREA.TXT el área de un círculo cuyo radio es un número aleatorio entero entre 2 y 27 inclusive
- c) mostrar por pantalla, en color azul, un mensaje indicando si un número entero ingresado por el usuario es múltiplo de 3 y 5.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>

int main()
{
    int opcion;
    float largo, ancho, areaRectangulo, perimetroRectangulo;
    int radio;
    float areaCirculo;
    int numEntero;
    printf("1 Area y perimetro de un rectangulo\n");
    printf("2 Area de un circulo\n");
    printf("3 Multiplo de 3 y 5\n");
    printf("Ingrese opcion (1, 2 o 3) :");
    scanf("%i",&opcion);
    switch(opcion)
    {
        case 1:
            printf("\nIngrese el largo : ");
            scanf("%f",&largo);
            printf("\nIngrese el ancho : ");
            scanf("%f",&ancho);
```

```

    areaRectangulo=largo*ancho;
    perimetroRectangulo=2*(largo+ancho);
    system("color F2");
    printf("\nEl area del rectangulo de largo %.2f y ancho %.2f",largo,ancho);
    printf(" es %.2f\n", areaRectangulo);
    printf("\nEl perimetro del rectangulo de largo %.2f y ancho %.2f",largo,ancho);
    printf(" es %.2f\n", perimetroRectangulo);
    break;
case 2:
    srand(time(NULL));
    radio=rand()%26+2;
    areaCirculo=M_PI*pow(radio,2.);

    FILE *archExt;
    archExt=fopen("c:/AREA.TXT","w");
    fprintf(archExt,"\nEl area del circulo de radio %i es %.2f\n",radio,areaCirculo);
    fclose(archExt);
    break;
case 3:
    printf("\nIngrese un numero entero : ");
    scanf("%i",&numEntero);
    if ((numEntero%3==0) && (numEntero%5==0))
    {
        system("color F1");
        printf("\nEl numero %i es multiplo de 3 y 5\n",numEntero);
    }
}

system("PAUSE");
return 0;
}

```

Salida1

```

1 Area y perimetro de un rectangulo
2 Area de un circulo
3 Multiplo de 3 y 5
Ingrese opcion <1, 2 o 3> :1
Ingrese el largo : 3
Ingrese el ancho : 3
El area del rectangulo de largo 3.00 y ancho 3.00 es 9.00
El perimetro del rectangulo de largo 3.00 y ancho 3.00 es 12.00
Presione una tecla para continuar . . .

```

Salida2

```
1 Area y perimetro de un rectangulo
2 Area de un circulo
3 Multiplo de 3 y 5
Ingrese opcion <1, 2 o 3> :3
Ingrese un numero entero : 15
El numero 15 es multiplo de 3 y 5
Presione una tecla para continuar . . .
```

Salida3

```
1 Area y perimetro de un rectangulo
2 Area de un circulo
3 Multiplo de 3 y 5
Ingrese opcion <1, 2 o 3> :2
Presione una tecla para continuar . . .
```

Observación:

Si observamos lo que aparece en la Salida3, notamos que el resultado no se visualiza en pantalla, porque este ha sido enviado al archivo de texto AREA.TXT que está localizado en el directorio raíz del disco c:

21

Escriba un programa en C que calcule, muestre por pantalla y guarde en un archivo de texto, en formato matricial, la inversa de una matriz no singular de orden 2 ingresada por el usuario. Para calcular el determinante debe usar una función de nombre Det.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

float Det(float a[2][2]);
float Det(float a[2][2])
{
    return(a[0][0]*a[1][1]-a[0][1]*a[1][0]);
}
int main()
{
    FILE *archtexto;
    archtexto=fopen("c:/tmp/inversa.txt","w");
    int i, j, n=2;
    float a[n][n], ainv[n][n];
    for(i=0;i<=n-1;i++)
        for(j=0;j<=n-1;j++)
            {
                printf("\nIngrese a[%i][%i] = ",i+1,j+1);
                scanf("%f",&a[i][j]);
            }
    ainv[0][0]=1/Det(a)*a[1][1];
    ainv[1][0]=-1/Det(a)*a[1][0];
    ainv[0][1]=-1/Det(a)*a[0][1];
    ainv[1][1]=1/Det(a)*a[0][0];
    for(i=0;i<=n-1;i++)
        {
            fprintf(archtexto,"\n");
            for(j=0;j<=n-1;j++)
                fprintf(archtexto,"\t%0.2f",ainv[i][j]);
        }
    fclose(archtexto);
    system("PAUSE");
    return 0;
}
```

Salida1

```
Ingrese a[1][1] = 1
Ingrese a[1][2] = 2
Ingrese a[2][1] = 3
Ingrese a[2][2] = 4
Presione una tecla para continuar . . .
```

Archivo inversa.txt

```
-2.00  1.00
1.50   -0.50
```

Salida2

```
Ingrese a[1][1] = 2
Ingrese a[1][2] = 1
Ingrese a[2][1] = 2
Ingrese a[2][2] = 1
Presione una tecla para continuar . . .
```

Archivo inversa.txt

```
1.#]   -1.#]
-1.#]   1.#]
```

Salida3

```
Ingrese a[1][1] = -1
Ingrese a[1][2] = 0.5
Ingrese a[2][1] = 3
Ingrese a[2][2] = 1.37
Presione una tecla para continuar . . .
```

Archivo inversa.txt

```
-0.48  0.17
1.05   0.35
```

Observaciones:

1) La Salida1 junto con el archivo inversa.txt muestran que

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1} = \begin{bmatrix} -2 & 1 \\ \frac{3}{2} & -\frac{1}{2} \end{bmatrix}$$

2) La Salida2 muestra un archivo inversa.txt con caracteres "raros", esto indica que la inversa de la matriz $\begin{bmatrix} 2 & 1 \\ 2 & 1 \end{bmatrix}$ no existe.

Esto nos dice que, a pesar que el enunciado solicita una matriz no singular, el usuario puede ingresar los números que desee.

3) La Salida3 junto con el archivo inversa.txt muestran que

$$\begin{bmatrix} -1 & 0.5 \\ 3 & 1.37 \end{bmatrix}^{-1} = \begin{bmatrix} -0.48 & 0.17 \\ 1.05 & 0.35 \end{bmatrix}$$

4) Este programa se puede mejorar, analizando si el determinante de la matriz ingresada es distinto de cero, para luego calcular su inversa.

Usando la orden switch, escriba un programa en C que muestre por pantalla un menú que permita al usuario elegir una de las opciones 1) ó 2) siguientes:

1) Guardar en un archivo externo, la traza de una matriz de orden k cuyos elementos, en el intervalo $[-15.2; 14.6]$, son ingresados por el usuario. El valor de k es un número entero mayor que 2 ingresado por el usuario. Debe implementar una función traza.

2) Muestre por pantalla la suma de las primeras m componentes de un vector de n componentes, con $m \leq n$. El usuario es quien ingresa el valor de m y las componentes del vector, y el valor de n es un número que se obtiene desde el archivo num.txt.

Solución:

Código Fuente

```
#include <stdio.h>
#include <stdlib.h>

float traza(int k, float a[k][k]);

float traza(int k, float a[k][k])
{
    int i, SumaDiagPrinc=0.0;
    for(i=0;i<=k-1;i++)
        SumaDiagPrinc=SumaDiagPrinc+a[i][i];
    return SumaDiagPrinc;
}

int main()
{
    int opcion;
    FILE *lee, *escribe;
    printf("\n1 Traza\n");
```



```

printf("\n2 Suma componentes\n");
printf("\n\nIngrese opci%c\n (1 %c 2) = ",162,162);
scanf("%i",&opcion);
switch(opcion)
{
case 1:
    {
    int k, i, j;
    do
    {
    printf("\n k = ");
    scanf("%i",&k);
    }
    while(k<=2);
    float a[k][k];
    for(i=0;i<=k-1;i++)
    for(j=0;j<=k-1;j++)
    {
    do
    {
    printf("a[%i][%i]= ",i+1,j+1);
    scanf("%f",&a[i][j]);
    }
    while(a[i][j]<-15.2 || a[i][j]>14.6);
    }
    escribe=fopen("e:/traza.txt","w");
    fprintf(escribe,"\nTraza = %0.1f\n",traza(k,a));
    fclose(escribe);
    break;
    }
case 2:
    {
    int m, n, i;
    float suma=0.0;
    lee=fopen("e:/num.txt","r");
    fscanf(lee,"%i",&n);
    float v[n];
    for(i=0;i<=n-1;i++)
    {
    printf("v[%i] = ",i+1);

```

```

        scanf("%f",&v[i]);
    }
do
{
    printf("\nIngrese m : ");
    scanf("%i",&m);
}
while(m>n);
for(i=0;i<=m-1;i++)
    suma=suma+v[i];
printf("\nLa suma es %0.2f\n\n",suma);
fclose(lee);
break;
}
}
printf("\n");
system("PAUSE");
return 0;
} 

```

Salida1

```

1 Traza
2 Suma componentes

Ingrese opción <1 ó 2> = 1

k = 3
a[1][1]= 1
a[1][2]= 2
a[1][3]= 3
a[2][1]= 4
a[2][2]= 5
a[2][3]= 6
a[3][1]= 7
a[3][2]= 8
a[3][3]= 9

Presione una tecla para continuar . . .

```

Archivo traza.txt

```
Traza = 15.0
```

Salida2

```
1 Traza
2 Suma componentes

Ingrese opción <1 ó 2> = 2
v[1] = 1
v[2] = 2
v[3] = 3

Ingrese n : 3
La suma es 6.00

Presione una tecla para continuar . . .
```

Archivo num.txt

```
3
```

Salida3

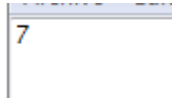
```
1 Traza
2 Suma componentes

Ingrese opción <1 ó 2> = 2
v[1] = -1
v[2] = 2
v[3] = 3
v[4] = 8
v[5] = 0
v[6] = 4
v[7] = -5

Ingrese n : 5
La suma es 12.00

Presione una tecla para continuar . . .
```

Archivo num.txt



Observaciones:

1) La sentencia

```
FILE *lee, *escribe;
```

se usó para declarar dos archivos, uno de lectura (lee) y otro de escritura (escribe)

2) La sentencia

```
escribe=fopen("e:/traza.txt","w");
```

indica que el archivo donde se guarda el valor de la traza de la matriz ingresada por el usuario, posee el nombre traza.txt, y además la unidad e: indica, en este caso particular, que ha sido guardado en un pendrive.

BIBLIOGRAFÍA

[1] Crandall, Richard E. *"Aplicaciones de Pascal en Ciencias"*. Editorial Limusa, Primera edición. 1988

[2] Waite, Mitchell; Prata, Stephen; Martin, Donald. *"Programación en C"*. Anaya Multimedia, Segunda edición. 1988

[3] Grossman, Stanley. *"Álgebra Lineal con Aplicaciones"*. Mc Graw Hill, 1992