

UNIVERSIDAD TECNOLÓGICA DE PEREIRA
PROGRAMA DE TECNOLOGÍA ELÉCTRICA
PROGRAMACIÓN TE243
TALLER: FUNCIONES QUE UTILIZA UN MENÚ

GRUPO: _____

FUNCIONES EN DEV C++.

ING: JOSÉ NORBEY SÁNCHEZ F.

FECHA: _____

Funciones en Lenguaje C.

La modularización, es una técnica usada por los programadores para hacer sus códigos más cortos, ya que consiste en reducir un gran problema complejo, en pequeños problemitas más sencillos, concentrándose en la solución por separado, de cada uno de ellos.

En C, se conocen como funciones aquellos trozos de códigos utilizados para dividir un programa con el objetivo que, cada bloque realice una tarea determinada.

En las funciones juegan un papel muy importante las variables, ya que como se ha dicho estas pueden ser locales o globales.

Variables Globales: Estas se crean durante toda la ejecución del programa, y son globales, ya que pueden ser llamadas, leídas, modificadas, etc; desde cualquier función. Se definen antes del main().

Variables Locales: Estas, pueden ser utilizadas únicamente en la función que hayan sido declaradas.

La sintaxis de una función es la siguiente:

```
Tipo_de_datos nombre_de_la_funcion(tipo y nombre de argumentos)  
{  
    acciones  
}
```

donde:

Tipo_de_datos: es el tipo de dato que devolverá esa función, que puede ser real, entera, o tipo void (es decir que no devolverá ningún valor).

Nombre_de_la_funcion: es el identificador que le damos a nuestra función, la cual debe cumplir las reglas que definimos en un principio para los identificadores.

Tipo y nombre de argumentos: son los parámetros que recibe la función. Los argumentos de una función no son más que variables locales que reciben un valor. Este valor se lo enviamos al hacer la llamada a la función. Pueden existir funciones que no reciban argumentos.

Acciones: constituye el conjunto de acciones, de sentencias que cumplirá la función, cuando sea ejecutada. Entre ellas están:

- Asignaciones
- Lecturas
- Impresiones
- Cálculos, etc

Una función, termina con la llave de cerrar, pero antes de esta llave, debemos colocarle la instrucción `return`, con la cual devolverá un valor específico. Es necesario recalcar que si la función no devuelve ningún valor, es decir, es tipo `void`, no tiene que ir la sentencia `return`, ya que de lo contrario, nos dará un error.

¿Cómo es que funcionan los Subprogramas?

A menudo, se utiliza el adjetivo de “Subprogramas”, para referirse a las funciones, así que, el lector debe familiarizarse también con este término.

Los subprogramas se comunican con el programa principal, que es el que contiene a las funciones, mediante parámetros, que estos pueden ser: **Parámetros Formales** y **Parámetros Actuales**.

Los parámetros se tratan como variables locales y los valores iniciales se proporcionan copiando los valores de los correspondientes argumentos.

Los **parámetros formales-Locales** de una función reciben como iniciales los valores de los **parámetros actuales** y con ellos se ejecutan las acciones descritas en el subprograma.

Funciones definidas por el usuario en C

Una función, como ya se ha dicho, es un bloque de códigos dentro del programa que se encargan de realizar una tarea determinada. Por lo tanto un programa en C, debe constar de una o más funciones, y por supuesto no puede faltar la función principal `main()`.

Un viejo adagio dice: **divide y vencerás**, lo cual se acopla perfectamente cuando tenemos un programa que es bastante grande; podemos separarlos en pequeños subprogramas (funciones), y concentrarnos en la solución por separados de cada uno de ellos y así resolver grandes problemas, en unos cuantos problemitas más pequeños.

Si un programa, está constituido por más de una función, las llamadas a la misma, pueden realizarse desde cualquier parte del programa, y la definición de ellas debe ser independiente unas de otras. Por lo tanto sería un grave error el tratar de definir una función dentro de otra.

Una función puede ser llamada desde cualquier parte del programa no sólo una vez, y cuando es llamada, empieza a ejecutar las acciones que están escritas en código.

Ejemplos de funciones:

- 1- Funciones que no devuelven ningún valor**
- 2- Funciones que devuelven un valor entero**
- 3- Funciones que devuelven un valor Real**
- 4- Funciones combinadas**
- 5- Funciones en las que usamos Menú.**

1- Funciones que no devuelven ningún valor.

Cómo se ha dicho las funciones pueden o no devolver algún valor, este tipo de funciones son las más sencillas, ya que cuando se llama la función, esta realiza lecturas, asignaciones, cálculos o impresiones, finaliza la ejecución de la función y el programa continúa normalmente.

Ejemplo 1: diseñe un programa que dados dos números enteros determine en una función la suma y en otra función cuál de ellos es mayor, menor o iguales.

/*Ejemplo 1, funciones que no devuelven un valor*/

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
void suma (int a, int b);      /*declaración de la función suma*/
void mayor (int a, int b);    /*declaración de la función mayor, menor o iguales*/

main()

{
    /*inicio del programa principal*/
    system("COLOR f1");
    int a, b;
    printf("Ingrese el valor de A:\n");
    scanf("%d", &a);
    printf("Ingrese el valor de B:\n");
    scanf("%d", &b);
    suma(a,b);                /*llamando cada una de las funciones*/
    mayor(a,b);               /*unicamente va el nombre de la funcion y de los parametros*/
    getch();
}
/*fin del programa principal*/

void suma(int a, int b)      /*definición de la función suma*/
{
    /*Abrimos llaves al inicio de la definición*/
```

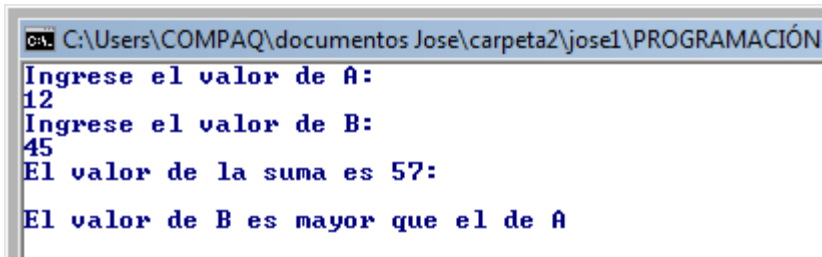
```

int sum;                /*Declaración de las variables locales*/
sum=a+b;
printf("El valor de la suma es %d:\n\n", sum);
}                        /*Fin de la función suma*/

void mayor(int a, int b) /*definición de la función mayor, menor o igual*/
{
    if(a==b)
        printf("Los dos números son iguales\n\n");
    else
    {
        if(a>b)
            printf("El valor de A es mayor que el de B\n\n");
        else
            printf("El valor de B es mayor que el de A\n\n");
    }
}                        /*fin de la funcion mayor, menor o igual*/

```

Salida:



```

C:\Users\COMPAQ\documentos Jose\carpeta2\jose1\PROGRAMACIÓN
Ingrese el valor de A:
12
Ingrese el valor de B:
45
El valor de la suma es 57:
El valor de B es mayor que el de A

```

2- Funciones que devuelven un valor entero

Las funciones que devuelven algún valor, se les llama **PROTOTIPOS DE FUNCIONES**. Antes de usar una función C debe tener conocimiento acerca del tipo de dato que regresara y el tipo de los parámetros que la función espera.

El estándar ANSI de C introdujo una nueva (mejor) forma de hacer lo anterior respecto a las versiones previas de C.

La importancia de usar prototipos de funciones es la siguiente:

- Se hace el código más estructurado y por lo tanto, más fácil de leer.
- Se permite al compilador de C revisar la sintaxis de las funciones llamadas.

Lo anterior es un hecho, dependiendo del alcance de la función. Básicamente si una función ha sido definida antes de que sea usada (o llamada), entonces se puede usar la función sin problemas.

Si no es así, entonces la función se debe declarar. La declaración simplemente maneja el tipo de dato que la función regresa y el tipo de parámetros usados por la función.

Es una práctica usual y conveniente escribir el prototipo de todas las funciones al principio del programa, sin embargo esto no es estrictamente necesario.

Para declarar un prototipo de una función se indicara el tipo de dato que regresará la función, el nombre de la función y entre paréntesis la lista del tipo de los parámetros de acuerdo al orden que aparecen en la definición de la función. Por ejemplo:

int longitud(int n); Lo anterior declara una función llamada longitud que regresa un valor entero y acepta otro valor entero como parámetro.

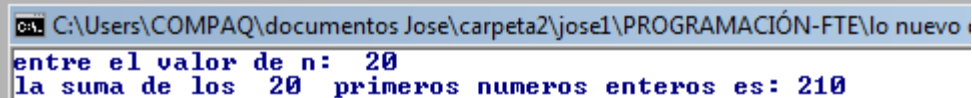
Ejemplo 2 : suma de n números enteros, donde n entra por teclado

```
/*Ejemplo 2, funciones que devuelve un valor entero*/
#include<conio.h>
#include<stdio.h>
#include <stdlib.h>
int suma(int n);           // Declaro la función antes de main y debe llevar (;)

main()
{
    system("COLOR f1");
    int sum,n;
    printf("entre el valor de n: " );
    scanf("%d",&n);
    printf("la suma de los %d primeros numeros enteros es: %d",n,suma(n));
    getch();
}

int suma(int n)           // definición de la función
{
    int i,sum=0;
    for(i=1; i<=n; i++)
        sum+=i;
    return(sum);
}
```

Salida:



```
C:\Users\COMPAQ\documentos Jose\carpeta2\jose1\PROGRAMACIÓN-FTE\lo nuevo
entre el valor de n: 20
la suma de los 20 primeros numeros enteros es: 210
```

3 - Funciones que Devuelven un Valor Real

Antes que nada, trataremos las funciones predefinidas en C. Ya que C, posee ciertas funciones que nos ayudan hacer nuestros programas más fáciles y utilizar menos código.

El lenguaje c, cuenta con una serie de funciones de bibliotecas que realizan operaciones y cálculos de uso frecuente.

Para acceder a una función, se realiza mediante el nombre seguido de los argumentos que le servirán a la función a realizar la tarea específica.

Nombre(arg1, arg2,...argn);

Ejemplo 3: Se desea conocer el resultado de las siguientes operaciones:

$\sqrt{a+b}$

$|a-b|$

Las variables a y b, son de tipo real, y pueden ser positivas o negativas.

/*Ejemplo 3, funciones que devuelven un valor real*/

#include <stdio.h>

#include <conio.h>

#include <math.h>

#include <stdlib.h>

double raiz(float a, float b); /*declaro la function raiz*/

double valor_absoluto(float a, float b); /*declaro la function valor_absoluto*/

main()

{

system("COLOR f1"); /*system pertenece a la libreria stdlib*/

float a, b;

printf("\n\n\t\tBIENVENIDO\n\n");

printf("Ingrese el valor de A, por favor:\n");

scanf("%f", &a);

printf("Ahora el valor de B:\n");

scanf("%f", &b);

printf("\n\nEl resultado de la raiz cuadrada de: %.2f + %.2f es: %.2f\n\n", a,b,raiz(a,b));

printf("\n\nEl valor absoluto entre |%.2f-%.2f| es igual a: %.2f\n\n",

a,b,valor_absoluto(a,b));

getch();

return 0;

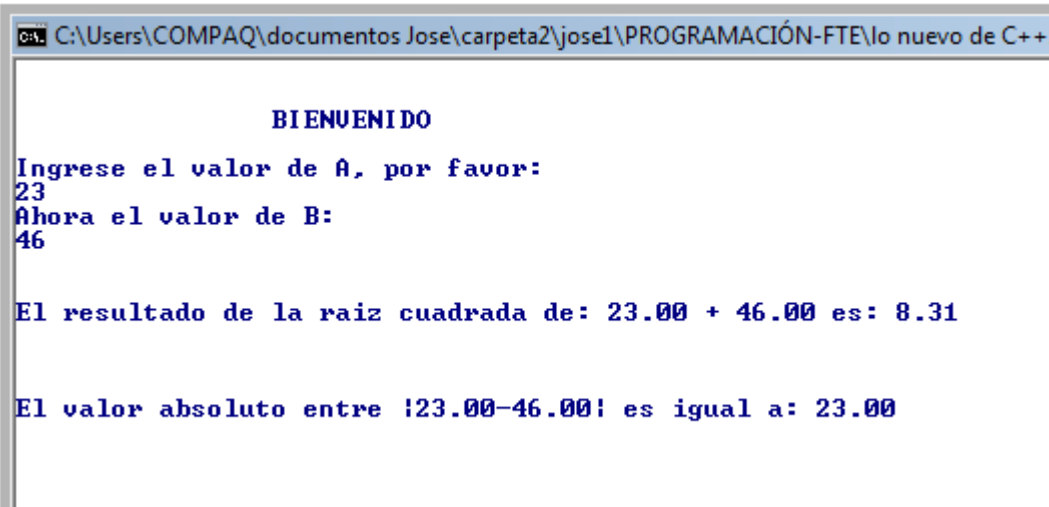
}

```

double raiz(float a, float b)          /*definicion de la funcion que devuelve un valor real*/
{
    float x;
    double y;
    x=a+b;
    y=sqrt(x);                         /* sqrt pertenece a la librería math*/
    return (y);                       /*valor real devuelto*/
}
double valor_absoluto(float a, float b) /*definicion de la funcion que devuelve un valor real*/
{
    float x;
    double y;
    x=a-b;
    y=fabs(x);                         /* fabs pertenece a la librería math*/
    return (y);                       /*valor real devuelto*/
}

```

Salida:



```

C:\Users\COMPAQ\documentos Jose\carpeta2\jose1\PROGRAMACIÓN-FTE\lo nuevo de C++

          BIENVENIDO
Ingrese el valor de A, por favor:
23
Ahora el valor de B:
46

El resultado de la raiz cuadrada de: 23.00 + 46.00 es: 8.31

El valor absoluto entre !23.00-46.00! es igual a: 23.00

```

4- Funciones Combinadas

A continuación veremos un ejemplo de un programa en el cual utilizamos dos funciones de diferente tipo de dato.

Ejemplo 4: programa que calcula el valor del número e, según el número de datos utilizando **dos ciclos do while**, uno para entrar el numero de términos y dar la respuesta del numero e y otro ciclo do while para finalizar el programa.

```

/*Ejemplo 4, funciones combinadas, una que devuelve un dato y otra que no*/
#include <stdio.h>
#include <conio.h>

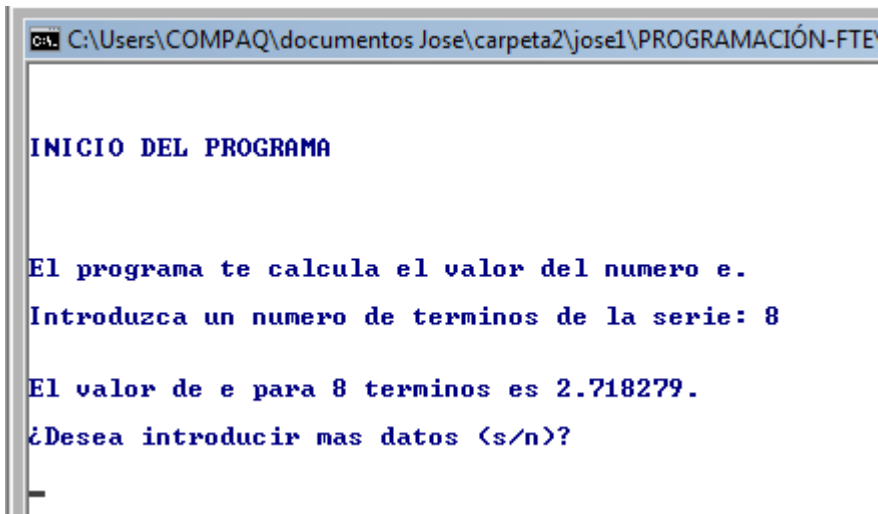
```

[illegible]

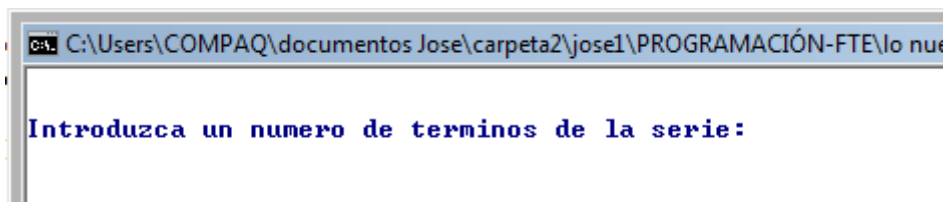

```
    printf("\n\nEl valor de e para %d terminos es %f.",numero,e);  
}
```

```
double factorial(int dato)          /*definicion de la funcion que devuelve un valor */  
{  
    int i;  
    double resultado=1;  
  
    for (i=dato;i>0;i--)  
    {  
        resultado=resultado*i;  
    }  
    return resultado;              /*valor devuelto*/  
}
```

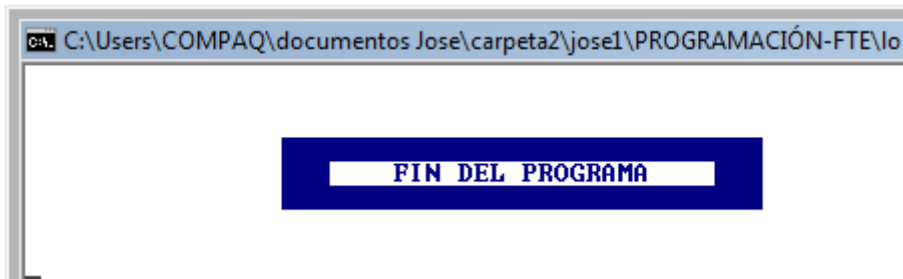
Salida:



```
C:\Users\COMPAQ\documentos Jose\carpeta2\jose1\PROGRAMACIÓN-FTE  
  
INICIO DEL PROGRAMA  
  
El programa te calcula el valor del numero e.  
Introduzca un numero de terminos de la serie: 8  
  
El valor de e para 8 terminos es 2.718279.  
¿Desea introducir mas datos (s/n)?  
_
```



```
C:\Users\COMPAQ\documentos Jose\carpeta2\jose1\PROGRAMACIÓN-FTE\lo nu  
  
Introduzca un numero de terminos de la serie:
```



```
C:\Users\COMPAQ\documentos Jose\carpeta2\jose1\PROGRAMACIÓN-FTE\lo  
  
FIN DEL PROGRAMA
```

5-Funciones en las que usamos Menú.

En la práctica, muchas veces debemos diseñar programas, que nos permitan elegir la acción o acciones a realizar, es decir haciendo uso de un menú. El cual, no es más ni menos que la aplicación de un selector múltiple. Un switch case que utiliza funciones.

Ejemplo 5: Diseñar un programa de funciones que utiliza un menú con 4 opciones. Este ejemplo sirve de base para el trabajo final de c.

/*Ejemplo 5, funciones en las que usamos menú */

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

```
#define MENSAJE printf("\n\n<<<<PRESIONE UNA TECLA PARA VOLVER A MENU");getch();
#define SI printf("\n\n<<<<SI...ENTONCES PULSA [S]");
#define NO printf("\n\n<<<<NO...ENTONCES PULSA UNA TECLA PARA VOLVER AL MENU");
```

```
double raiz(float a, float b);
double valor_absoluto(float a, float b);
int suma(int n);                /*Declaro todas las funciones antes de main y deben
                                llevar (;)*/
```

/*FUNCION MENU: MUESTRA UN MENU EN PANTALLA*/

```
char menu()
{
    printf("\n\n\n\t\tBIENVENIDO/A\n\n");
    printf("\n\n\t\tMENU DE OPCIONES\n\n");
    printf("\n*****\n");
    printf("\n\t1-RAIZ CUADRADA DE DOS NUMEROS ENTEROS");
    printf("\n\t2-VALOR ABSOLUTO DE DOS NUMEROS ENTEROS");
    printf("\n\t3-SUMA DE LOS PRIMEROS NUMEROS ENTEROS");
    printf("\n\n\t4-SALIR");
    printf("\n\n\tESCOJA UNA OPCION");
    printf("\n*****\n");
}
```

/*INICIO DEL PROGRAMA PRINCIPAL*/

```

main()
{
    system("COLOR f1");           /*system pertenece a la libreria stdlib*/
    float a,b;                   /*declaracion de variables globales*/
    int n;
    char c,resp;

do
{
    do
    {
        system("CLS()");         /*limpia pantalla*/
        menu();
        c=getche();
        getch();
        switch(c)
        {
            case '1':system("CLS"); /*limpia pantalla*/
                printf("\n\t\tRAIZ CUADRADA DE DOS NUMEROS ENTEROS\n\n");
                printf("\nINGRESE EL VALOR DE A, POR FAVOR: ");
                scanf("%f", &a);
                printf("\nAHORA EL VALOR DE B: ");
                scanf("%f", &b);
                printf("\nEL RESULTADO DE LA RAIZ CUADRADA DE: %.2f + %.2f ES: %.2f\n\n",
                    a,b,raiz(a,b));
                MENSAJE;
                break;

            case '2':system("CLS"); /*limpia pantalla*/
                printf("\n\t\tVALOR ABSOLUTO DE DOS NUMEROS ENTEROS\n\n");
                printf("\nINGRESE EL VALOR DE A, POR FAVOR: ");
                scanf("%f", &a);
                printf("\nAHORA EL VALOR DE B: ");
                scanf("%f", &b);
                printf("\nEL VALOR ABSOLUTO ENTRE: |%.2f-%.2f| ES IGUAL A: %.2f\n\n",
                    a,b,valor_absoluto(a,b));
                MENSAJE;
                break;

            case '3':system("CLS"); /*limpia pantalla*/
                printf("\n\t\tSUMA DE LOS PRIMEROS NUMEROS ENTEROS \n\n");
                printf("\nENTRE EL VALOR DE N: " );

```



```

for(i=1; i<=n; i++)
    sum+=i;
    return(sum);                /*funcion que devuelve un valor*/
}

```

Salida:

```

C:\Users\COMPAQ\Desktop\menuconvoydaller.exe

      BIENVENIDO/A

      MENU DE OPCIONES

*****

      1-RAIZ CUADRADA DE DOS NUMEROS ENTEROS
      2-VALOR ABSOLUTO DE DOS NUMEROS ENTEROS
      3-SUMA DE LOS PRIMEROS NUMEROS ENTEROS
      4-SALIR

      ESCOJA UNA OPCION
*****

```

```

C:\Users\COMPAQ\Desktop\menuconvoydaller.exe

      RAZ CUADRADA DE DOS NUMEROS ENTEROS

INGRESE EL VALOR DE A, POR FAVOR: 23
AHORA EL VALOR DE B: 45
EL RESULTADO DE LA RAZ CUADRADA DE: 23.00 + 45.00 ES:8.25

<<<<PRESIONE UNA TECLA PARA VOLVER AL MENU_

```

```

C:\Users\COMPAQ\Desktop\menuconvoydaller.exe

      VALOR ABSOLUTO DE DOS NUMEROS ENTEROS

INGRESE EL VALOR DE A, POR FAVOR: 24
AHORA EL VALOR DE B: 78
EL VALOR ABSOLUTO ENTRE:124.00-78.00! ES IGUAL A:54.00

<<<<PRESIONE UNA TECLA PARA VOLVER AL MENU

```

```
C:\Users\COMPAQ\Desktop\menuconvoidtaller.exe

SUMA DE LOS PRIMEROS NUMEROS ENTEROS

ENTRE EL VALOR DE N: 20
LA SUMA DE LOS 20 PRIMEROS NUMEROS ENTEROS ES:210
<<<<PRESIONE UNA TECLA PARA VOLVER AL MENU_
```

```
C:\Users\COMPAQ\Desktop\menuconvoidtaller.exe

ERROR<<<<DEBES ELEGIR UNA OPCION VALIDA
QUIERES SALIR DEL PROGRAMA?
<<<<SI...ENTONCES PULSA [S]
<<<<NO...ENTONCES PULSA UNA TECLA PARA VOLVER AL MENU
```

Trabajo final de C:

Cada uno de los estudiantes debe diseñar un programa con funciones en las que usamos un menú de opciones como en el ejemplo 5 visto anteriormente.

El menú debe contener 6 opciones (ver lista de los 6 ejercicios que le corresponden según su código en la tabla de abajo); cada opción representa una función una vez llamada ejecutar el subprograma y mostrar resultados, además se debe tener la opción de seguir o salir del programa sin abortar. Preferiblemente nombrar el tipo de dato de la función con **void**.

Nota: me envían el trabajo por correo (insanchez@utp.edu.co) y lo sustentan individualmente. Plazo hasta semana muerta.

Lista de ejercicios:

1) Nota definitiva de programación. Utilizando la estructura if-then-else

Primer parcial vale	20%
Segundo parcial vale	25%
Tercer parcial vale	30%
Exámen final vale	25%

Promedio de nota ?

Si promedio de nota $\geq 3 \Rightarrow$ Ganó la materia, nota = ?

$2.5 < \text{promedio de nota} < 3 \Rightarrow$ Perdió la materia, nota =?

Promedio de nota $\leq 2.5 \Rightarrow$ Quedó en transición, nota =?

Nota: debe entrar datos por teclado, mostrar notas parciales, promedio y como queda el estudiante

2) Decir si un número real leído por teclado es positivo, negativo o cero. Utilizando la estructura if-then-else

3) Diseñar un programa que calcule las raíces de la ecuación cuadrática de la forma:
 $ax^2 + bx + c = 0$, utilizando la estructura if-then-else

4) Determinar el salario neto de un empleado de una compañía con base en el salario básico, bonificación, descuento y categoría. Considere la siguiente tabla:

categoría	bonificación	descuento
1	15%	4%
2	13%	3%
3	12%	2%
4	9.5%	0%

Utilizando la estructura if-then-else

5) Realizar un programa con el cual se puedan hacer las siguientes operaciones con 3 números. Utilizando la estructura if-then-else

- Calcular el promedio
- Hallar el mayor
- Hallar el menor
- Identificar cuántos pares e impares hay

6) Diseñar un programa utilizando la ley de ohm ($V=I \times R$), elija que desea calcular? V, I, o R. Utilizando la estructura if-then-else

7) Realizar un programa que dados dos números ejecute las 4 operaciones básicas (+, -, /, *)

- Suma
- resta
- multiplicación
- división

8) Diseñar un programa que calcule:

- área de un círculo
- longitud de un círculo

9) Diseñar un programa que calcule:

- el área de un rectángulo
- área de un triángulo rectángulo
- área de un trapecio

10) Hallar la suma y el promedio de los 20 primeros números enteros mayores que cero, utilizando la sentencia while

11) Hallar la suma total, suma de pares e impares entre 1 y n, donde n entra por teclado, utilizando la sentencia for

12) Calcular el cuadrado y el cubo de los 15 primeros números mayores que cero, utilizando la sentencia while

13) realizar un programa donde se digiten 7 números enteros y diga ¿Cuántos pares hay?, y muestre también los números digitados.

14) El siguiente es el menú de un restaurante de bocadillos. Diseñar un algoritmo capaz de leer el número de unidades consumidas de cada alimento ordenado y calcular la cuenta total.

- Bocadillo de jamón (\$250)
- Bocadillo de queso (\$200)
- Patatas fritas (\$100)
- Refresco (\$175)
- Cerveza (\$125)

15) Determinar la suma y el producto de los números pares entre 20 y 40 inclusive. Realizar el programa con la sentencia while

16) Diseñar un algoritmo que realice la siguiente conversión. Una temperatura dada en grados Centígrados (C) pasar a grados Fahrenheit (F) y viceversa, de acuerdo a la siguiente expresión

$$F = \frac{9}{5} \times C + 32$$

17) Realizar el algoritmo que permita calcular la suma de los números múltiplos de 5 comprendidos entre 1 y N, donde N entra por teclado con la sentencia while

18) Realice el algoritmo que permita calcular la suma de los números múltiplos de 3 comprendidos entre 1 y N, donde N entra por teclado con la sentencia for

19) Realizar un algoritmo que permita emitir una factura correspondiente a una compra de un artículo determinado del que se adquiere 1 o varias unidades, el IVA a aplicar es el 16%. Si el precio bruto es mayor a \$200.000.00 se debe realizar un descuento del 5%

20) Encontrar la media de n números positivos leídos por teclado, la media se expresa como:

$$Media = \frac{X_1 + X_2 + X_3 + \dots + X_n}{n}$$

21) Determinar la corriente que alimenta un circuito con n resistencias en serie. El valor de la tensión en voltios, el número de resistencias así como el valor de cada resistencia debe ser ingresado por teclado. Recordar: $V = I * R_{eq}$.

22) Un voltímetro envía señales de sus lecturas a un computador el cual debe mostrar en pantalla un mensaje de acuerdo a la lectura. Diseñe un programa que acepta una entrada de voltaje y muestra un mensaje en pantalla de la siguiente manera: Utilizando la estructura if-then-else

Voltaje	Mensaje
0-110	Bajo
110-120	Medio
120-200	Alto

23) Una Empresa de energía necesita un programa para facturar la energía consumida en la ciudad, según la siguiente tabla. El programa debe pedir los kwh (kilovatio hora) consumidos y entregar el valor a pagar según el estrato, utilizando la estructura if-then-else

Ítem	Estrato 1	Estrato 2	Estrato 3
Rango estrato	40 kwh	80 kwh	130 kwh
Precio (\$ / Kwh)	\$180	\$185	\$189

24) Realizar un Programa en C, empleando la sentencia iterativa Do While, que muestre por pantalla los números pares del 1 al 100

25) Realizar un Programa en C, empleando la sentencia iterativa for, que muestre por pantalla los primeros 30 números naturales, pero a la inversa del 30 al 1

26) Desarrollar un programa que capture un número, adicionalmente verifique si ese número ingresado está comprendido entre los siguientes rangos:

- Si es menor a 10
- Si es mayor a 10 y menor que 100
- Si es mayor a 100.

En cada caso emitir un mensaje correspondiente al mismo. Utilice la estructura if-then-Else

27) Suponga que un individuo desea invertir su capital en un banco y quiere saber cuánto dinero ganará después de un año si el banco paga a razón de 2% mensual.

28) En un hospital existen tres áreas: Ginecología, Pediatría y Traumatología. El presupuesto anual del hospital se reparte conforme a la siguiente tabla:

Área	Porcentaje del presupuesto
Ginecología	40%
Traumatología	35%
Pediatría	25%

Obtener la cantidad de dinero que recibirá cada área, para cualquier monto presupuestal. Utilizando la estructura if-then-else

29) Calcular el número de pulsaciones que una persona debe tener por cada 10 segundos de ejercicio, si la formula es:

$$num_{pulsaciones} = (220 - edad) / 10$$

30) calcular el mínimo de un vector de 10 números y su posición utilizando la sentencia for. Debe mostrar el vector y su posición

31) calcular e imprimir la suma de los números impares comprendidos entre 1 y m, donde m entra por teclado. Realizar el algoritmo con la sentencia while

32) digite un número entero y calcule con la sentencia for:

- la suma de los primeros números consecutivos
- la suma de pares consecutivos
- la suma de impares consecutivos

ESTUDIANTES PRIMER SEMESTRE DE 2011

No.	Prog	Código	calificación	20%	Número de ejercicios
1	22	1088270448			1 13 18 12 19 6
2	22	1088293243			2 8 25 24 23 16
3	22	1088290924			3 7 30 31 22 19
4	22	1088272873			4 21 32 9 26 17
5	22	1088296047			5 20 11 10 9 4
6	22	1088293612			6 16 18 15 14 23
7	22	1117518362			22 19 30 17 13 2
8	22	1088009726			23 9 32 24 20 2
9	22	15924405			26 27 11 10 29 1
10	22	91121826440			28 14 18 12 7 2
11	22	1088297918			1 29 25 10 28 7
12	22	1093219864			2 27 30 15 26 8
13	22	1088285422			3 21 32 12 23 9
14	22	1088289806			4 20 11 17 22 14
15	22	93102708804			5 19 18 31 28 16
16	22	1088255894			6 16 25 10 23 19
17	22	1088265526			22 14 30 15 4 29
18	22	1088008184			23 13 32 12 1 14