

TP 3 - Funciones



Objetivo

- * Identifique cada una de los componentes de las funciones y tipos de funciones.
- * Entienda cuando es útil y porque es conveniente dividir el código en funciones.
- * Reconozca y determine cuando es útil utilizar ***los parámetros por valor y/o por referencia.***
- * Reconozca y sepa cuándo es conveniente y cuando no utilizar funciones.
- * Aplique funciones correctamente en un lenguaje de programación.
- * Reconozca la importancia y diferencia entre ***variable local y global.***
- * Implemente adecuadamente los tipos de funciones en Lenguaje C.

¿Qué es una Función?

Una función es un segmento del programa que tiene **un nombre, es independiente, ejecuta una tarea específica** y que opcionalmente **regresa un valor al programa** que la llama.

Una función tiene nombre. Cada función tiene un nombre único. Con ese nombre, en cualquier otra parte del programa, se pueden ejecutar los enunciados contenidos en la función. A esto se le conoce como la **llamada de la función**.

Una función es independiente. Una función puede ejecutar su trabajo sin interferencia de, y sin interferir con, otras partes del programa.

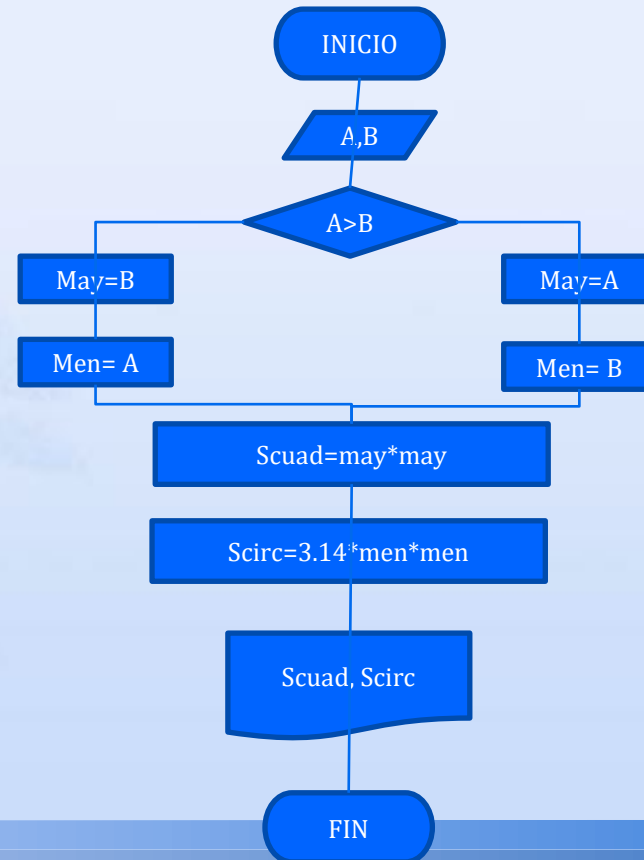
Una función ejecuta una tarea específica. Una tarea es un trabajo concreto que un programa debe ejecutar como parte de su operación general, como enviar una línea de texto a la impresora, mostrar un menú en pantalla, ordenar un arreglo o calcular una raíz cúbica.

Una función puede regresar un valor. Cuando es llamada la función se ejecutan los enunciados que contiene, y en caso de que se desee, pueden pasar información de regreso al programa que la llama.

El uso de funciones permite la descomposición y desarrollo modular.

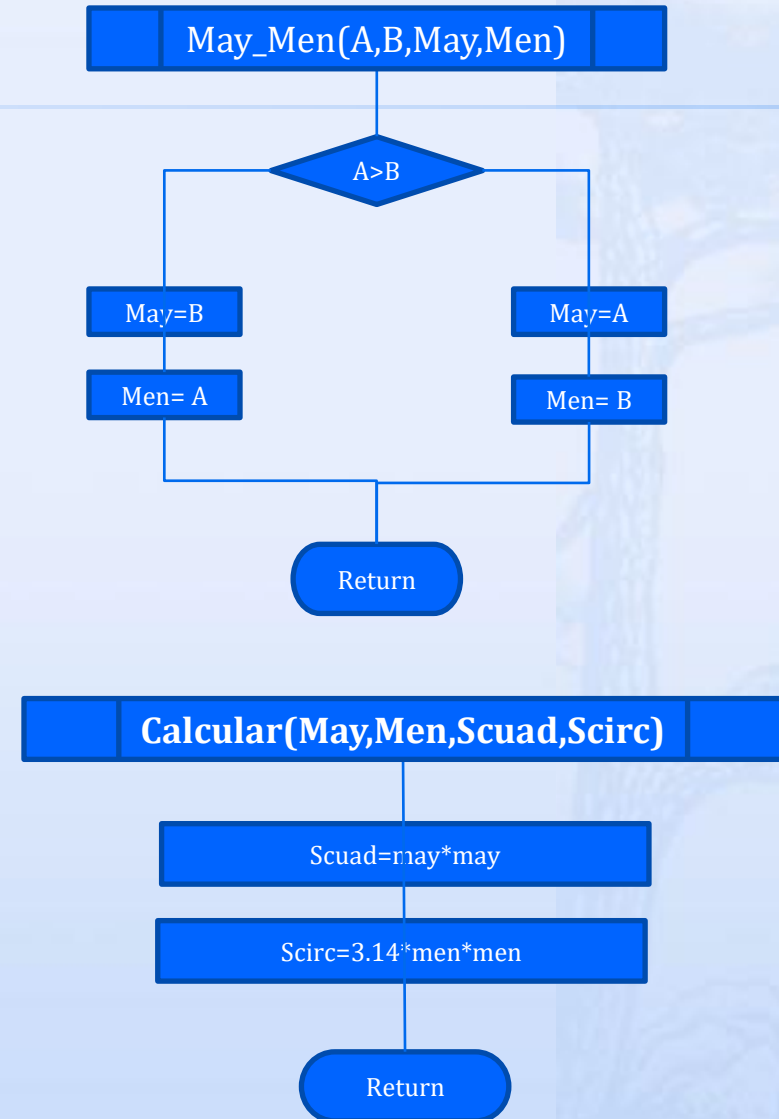
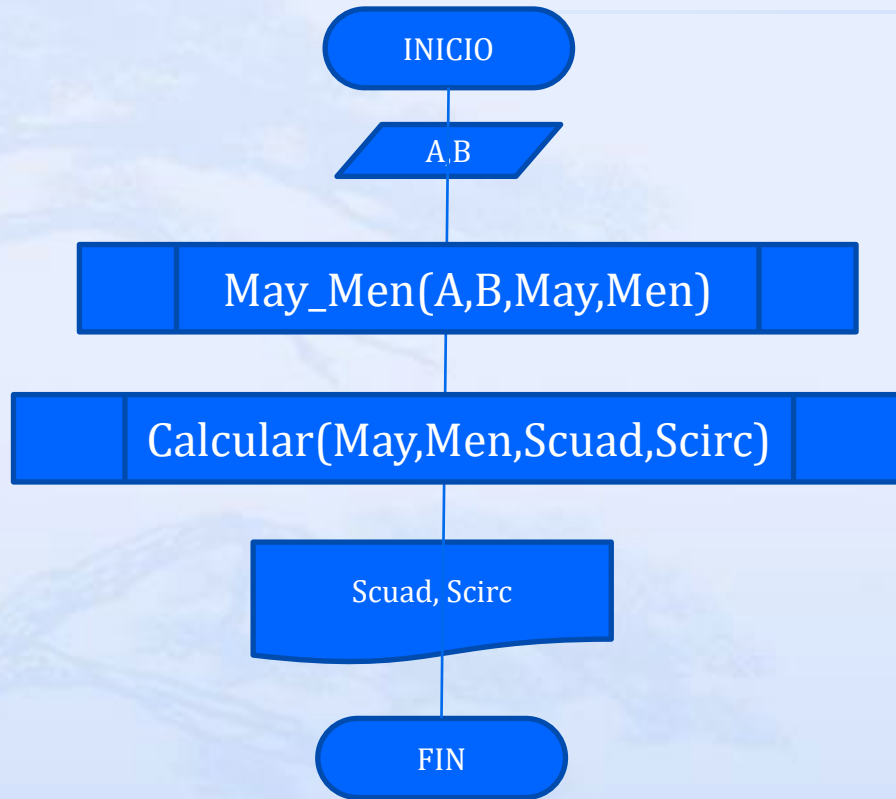
Permite dividir un programa en componentes más pequeños y además puede ser **reutilizado**.

Ej. Se conocen dos números distintos. Calcular la superficie de un cuadrado, suponiendo como lado del mismo al mayor de los números dados y la superficie de un círculo suponiendo como radio del mismo al menor de los números dados.



```
# include <stdio.h>
#include <stdlib.h>
float a,b,men,may,scuad,scirc;
main()
{
    clrscr();
    printf("Ingrese el primer valor: ");
    scanf("%f",&a);
    printf("Ingrese el segundo valor: ");
    scanf("%f",&b);
    if (a>b)
    {
        may=a;
        men=b;
    }
    else
    {
        may=b;
        men=a;
    }
    scuad=may*may;
    scirc=3.14*men*men;
    printf("\nLa superficie del cuadrado es: %f",scuad);
    printf("\nLa superficie del circulo es: %f",scirc);
    system ("PAUSE");
}
```

Problema del área del cuadrado y círculo pero usando subrutina



Clasificación de Funciones

- ✧ En todo lenguaje existen 2 tipos de funciones:
 - ✧ Funciones **con tipo** (o con retorno)
 - ✧ Funciones **sin tipo** (o sin retorno)

Funciones con tipo

Son aquellas que devuelven un valor como resultado de la acción que realizan, de forma tal que ese valor puede volver a usarse en alguna otra operación. La función **sqrt** es de esta clase, y puede usarse, por ejemplo, en cualquiera de las formas siguientes:

```
Y=sqrt(x);  
printf("%f",sqrt(x));  
Y=2*sqrt(x);  
Y=sqrt(sqrt(x));
```


Funciones sin tipo

Son aquellas que realizan alguna acción pero no retornan valor alguno como resultado de la misma. Un ejemplo es la función **clrscr()** que limpia la pantalla. Estas funciones no pueden ser utilizada en las formas que se indicaron para las funciones con retorno. Para invocarlas, simplemente se escribe su nombre y se pasan los parámetros que sean necesarios.

Definición de una función

```
tipo nombre_func (tipo1 arg1, ..., tipoN argN)
{
/* CUERPO DE LA FUNCION */
}
```

```
float promedio (int a, int b)
{
/* CUERPO DE LA FUNCION */
}
```

- * Los argumentos se denominan parámetros formales.
- * La función devuelve un valor de tipo de dato **tipo**
 - * Si se omite tipo se considera que devuelve un **int**
 - * Si no devuelve ningún tipo => **void** (significa vacío)
 - * Si no tiene argumentos => **void**

void explicacion(void)

- * Entre llaves se encuentra el cuerpo de la función (igual que main()).
- * La sentencia **return** finaliza la ejecución y devuelve un valor a la función que realizó la llamada.

return(expresion);

Paso de parámetros por referencia

- ✿ En este tipo de llamadas los argumentos contienen **direcciones de variables**.
- ✿ Mediante **&** podemos pasar direcciones de variables en lugar de valores.


TP 3 de FUNCIONES

1) ✎ Hacer un programa que simule un cajero automático con un saldo inicial de 10000 Pesos. Mi menú estaría formado de la siguiente manera.

Bienvenido a su cajero automatico

1. Consulta del Saldo
2. Ingresar dinero en cuenta
3. Retirar dinero de la cuenta
4. Salir

Elija su Opcion:

- 1)  Se ingresa una cantidad **N** de números entero de cinco dígitos de los cuales a medida que se ingresan se ira determinando si dicho número es palíndromo. Verificar que el valor ingresado sea de 5 dígitos.

SE PIDE

- a) Determinar si el número ingresado es palíndromo, *Usando una función sin tipo*. Su prototipo será el siguiente: **void nombreFuncion(parametro1, parametro2);**
- i) El parámetro1 recibe por valor el número ingresado por teclado.
 - ii) El parámetro2 es un parámetro por **referencia** el cual indicara
 - con un valor 1, si el número ingresado es un palíndromo.
 - Con un valor 0, si el numero ingresado NO es un palíndromo.
- b) En la función main() se mostrara un mensaje indicando si el numero ingresado es palíndromo o no.

TP 3 de FUNCIONES

2) ✎ Se ingresa un número entero X de cinco dígitos, validar la entrada.

- a) En la función main() ingresar un numero entero positivo.
- b) Determinar si el número ingresado es **palíndromo**, *Usando una función sin tipo. Usando parámetro por Valor transferir el número a la función y parámetro por referencia para saber si es palíndromo. En el caso de que sea palíndromo la variable tendrá el valor **1(unos)** o **0(cero)** si no lo es.*
- c) En la función main() se mostrara un mensaje indicando si el numero ingresado es palíndromo o no.

EJ. De Palindromo

Lavan esa base naval

4568654

X= 45645

X= 73637

Variables a usar

NI para el nro inverso
PE para la parte entera
R para el resto

Algoritmo

NI=0 (*inicializamos en 0 el Nro Inverso*)

PE=X (*asignamos el nro a analizar a la Parte Entera*)

R=5

PE=4564 (4 Digitos)

NI=NI*10+R=0*10+5=5

R=4

PE=456 (3 Digitos)

NI=5*10+4=54

R=6

PE=45 (2 Digitos)

NI=54*10+6=546

R=5

PE=4 (1 Dígito)

NI=546*10+5=5465

R=4

PE=0

NI=5465*10+4=54654

USAR UNA ESTRUCTURA WHILE

MIENTRAS LA Parte Entera sea DISTINTO DE cero

Se repite el algoritmo

$R = (PE \text{ MOD } 10);$
 $PE = PE/10;$
 $NI = NI*10 + R;$

Saca el Resto
Saca la Parte Entera
Construye el nro a la Inversa

INICIO

Ingresar el Nro a evaluar

Validar de que sea de 5 dígitos

Llamar a la función PALINDROMO enviando el nro como parámetro

Si la variable Pali = 1

Mostrar mensaje de ES PALINDROMO

CASO CONTRARIO

Mostrar mensaje NO ES PALINDROMO

FIN

FUNCION PALINDROMO

PE=Nro ingresado

NI=0

MIENTRAS PE sea distinto de cero HAGA

$R = (PE \% 10);$

$PE = PE / 10;$

$NI = NI * 10 + R;$

FIN DEL MIENTRAS

Si Nro Ingresado = NI

Pali=1

Caso Contrario

Pali=0

FIN DE LA FUNCION

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

```
void Palindromo(int XNum, int &EsPalindromo); //Prototipo de la función
```

```
main()
{
```

```
    int X, EsPalindromo;
```

```
    do{
```

```
        printf("Ingrese un Numero ENTERO de 5 cifras\n\n");
        scanf("%d",&X);
        if(X<10000 or X>99999) //Determina si el numero es de 5 cifras.
        {
```

```
            printf("Error el numero no es valido\n\n");
            system("PAUSE");
            system("CLS");
```

```
        }
```

```
    }while(X<10000 or X>99999);
```

```
    EsPalindromo=0;
```

```
    Palindromo(X, EsPalindromo); //Llamada a la función.
```

```
    if(EsPalindromo==1)
```

```
        printf("ES P A L I N D R O M O \n\n");
```

```
    else
```

```
        printf("NO ES P A L I N D R O M O\n\n");
```

```
    system("PAUSE");
```

```
}
```

```
void Palindromo(int XNum, int &EsPalindromo) //recibe un valor xNum y por referencia  
esPalindromo.
```

```
{
```

```
    int NI, PE, R;
```

```
    NI = 0;
```

```
    PE = XNum;
```

```
    R = 0;
```

```
    while(PE != 0) //Obtiene el palindromo
```

```
    {
```

```
        R = (PE%10);
```

```
        PE = PE/10;
```

```
        NI = NI*10 + R;
```

```
    } //Fin
```

```
    //Evalua si en inverso obtenido y el numero ingresado son iguales.
```

```
    if(XNum==NI)
```

```
        EsPalindromo = 1; //Cambia el valor de la variable declarada en main.
```

```
    else
```

```
        EsPalindromo = 0; //Cambia el valor de la variable declarada en main.
```

```
} //Final de la función que determina si un valor de 5 cifras, es palindromo.
```

CREAR NUESTRA LIBRERIA

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

int Palindromo(int XNum) //recibe un valor ENTERO DE 5 DIGITOS
{
    int NI, PE, R, EsPalidromo;
    NI = 0;
    PE = XNum;
    R = 0;
    while(PE != 0) //Obtiene el palindromo
    {
        R = (PE%10);
        PE = PE/10;
        NI = NI*10 + R;
    } //Fin
    //Evalua si en inverso obtenido y el numero ingresado son iguales.
    if(XNum==NI)
    {
        EsPalindromo = 1; //Cambia el valor de la variable declarada en main.
    }
    else
    {
        EsPalindromo = 0; //Cambia el valor de la variable declarada en main.
    }
    return(EsPalindromo);
} //Final de la función que determina si un valor de 5 cifras, es palindromo.
```


Una vez creada nuestra función debemos grabar con un nombre con la Extensión .h en la carpeta include de nuestro DEV C instalado.

Por ejemplo pali.h y cuando la invoquemos a la función debemos incluir la Librería creada.

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

main()
{
    int X, Palindro;

    do{
        printf("Ingrese un Numero ENTERO de 5 cifras\n\n");
        scanf("%d",&X);
        if(X<10000 or X>99999) //Determina si el numero es de 5 cifras.
        {
            printf("Error el numero no es valido\n\n");
            system("PAUSE");
            system("CLS");
        }
    }while(X<10000 or X>99999);
    Palindro=Palindromo(X); //Llamada a la función.
    if(Palindro==1)
        printf("ES P A L I N D R O M O \n\n");
    else
        printf("NO ES P A L I N D R O M O\n\n");
    system("PAUSE");
}
```

3)  Se ingresan dos números distintos y enteros,

SE PIDE:

- a) Calcular la superficie de un cuadrado. Suponiendo como lado del cuadrado al mayor valor de los números ingresados. Su resultado debe ser mostrado en la función main().
- b) Calcular la superficie de un círculo. Suponiendo como radio del mismo al menor de los números ingresado. Su resultado debe ser mostrado en la función main().

NOTA

*En este problema se ve la presencia de **tres funciones** definidas por el programador, para nuestro caso serán funciones **sin tipo**, cuyos prototipos son los siguientes:*

 Prototipo para, Calcular el menor y el mayor.

void CalMayMen (**int** &May, **int** &Men);


 Prototipo para, Calcular la superficie del cuadrado.

void AreaCuad(**int** May, **int** &SupCua);

 Prototipo para, Calcular la superficie del cilindro.

void AreaCilin(**int** Men, **int** &SupCil);


USO DE BIBLIOTECA PROPIA

- 1)  En este ejercicio se realizará la creación de varias funciones las cuales serán guardadas como una biblioteca (librería) para luego ser utilizadas en un programa.
 - a) Crea una librería que contenga las siguientes funciones de tipo: ⁽¹⁾Retorne el mayor valor de 3 valores cualquiera, ⁽²⁾Retorne el menor valor de 3 valores cualquiera, ⁽³⁾Retorne el promedio de 3 valores cualquiera, ⁽⁴⁾Retorne el resultado del producto de dos valores, ⁽⁵⁾ Retorne el resultado de la división de dos valores cualquiera. Tener en cuenta división en cero, ⁽⁶⁾Retorne el resultado de Sumar dos valores, ⁽⁷⁾Retorne el resultado de restar dos valores.
 - b) Una vez creadas las funciones, deberás guardar el archivo **nombre.h** , el nombre lo eliges según tu criterio, en la carpeta "**C:\Program Files (x86)\Dev-Cpp\MinGW32\include**" esto puede variar dependiendo de la versión del sistema operativo y/o del lugar donde fue instalado el DEV-C.
 - c) Ahora realiza un programa que permita el ingreso de 3 valores enteros.

SE PIDE: Usando nuestra biblioteca nos muestre:

- Cuál es el mayor valor y el menor valor de los valores ingresados.
- Cuál es el Promedio de los valores ingresado.
- El resultado de sumar el primer valor con el ultimo valor
- El resultado de restar el mayor valor ingresado, con el menor valor ingresado.
- El resultado de la división del mayor valor ingresado en el segundo valor ingresado.
- El resultado del producto de los 3 valores ingresados.

Uso de función RECURSIVA

- 1)  Desarrolla una función recursiva que permita obtener el producto de dos números (**a** y **b**) enteros positivos mayor a cero. Aplicando el concepto de “*sumar el número a la cantidad de veces que indica el numero b*”.

POR EJEMPLO: $a = 3$ y $b=5$; entonces sería $3+3+3+3+3 = 15$

*5 veces se suma a para obtener $a*b$*

POR EJEMPLO a=2 y b=3

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

int Producto(int a, int b);

main()
{
    int a, b;
    printf("Ingrese dos numeros enteros mayores a 0.\n");
    printf("Ingrese el primer numero: ");
    scanf("%d", &a);
    printf("Ingrese el segundo numero: ");
    scanf("%d", &b);
    printf("Producto: %d", Producto(a, b));
}

int Producto(int a, int b)
{ int pro;
    if (b==1)
        pro=a;
    else
        pro=(a+Producto(a, (b-1)));

    return pro;
}
```

```
Producto(2, 3)
{
    if (3==1)
        return 2;
    else
        return (2+Producto(2, 2));
```


2+4=6

```
Producto(2, 2)
{
    if (2==1)
        return 2;
    else
        return (2+Producto(2, 1));
```

2+2=4

```
Producto(2, 1)
{
    if (1==1)
        return 2;
    else
        return (2+Producto(2, 0));
```

2

- 2)  Desarrolla una función recursiva que permita dividir en forma entera un número positivo **a**, mayor a cero por otro número positivo **b**, menor que **a**, aplicando el principio de “*determinar cuántas veces cabe b en a aplicando restas sucesivas*”.

EJEMPLO: **a** = 6 y **b**=2; entonces sería $\overbrace{6-2=4}^{1\text{ vez}}$ $\overbrace{4-2=2}^{2\text{ veces}}$ $\overbrace{2-2=0}^{3\text{ veces}}$ **Respuesta 3.** O sea que $(6/2 = 3)$

*Se cuenta el número de veces que se resta **a-b** para obtener **a/b***

```
int division(int a, int b)
{
    int divisor;
    if ((a-b)<0)
    {
        divisor=0;
    }
    else
    {
        divisor=1 + division(a-b,b);
    }
    return divisor;
}
```



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

```
int division(int a, int b);
```

```
main()
{
    int numa=0,numb=0,resultado=0;
    do
    {
        printf("\nIngrese un valor para a mayor a 0(dividendo): ");
        scanf("%d",&numa);
        if(numa<0)
        {
            printf("\nEl numero tiene que ser mayor que 0. Ingrese nuevamente\n");
            system("pause");
        }
    }while(numa<0);
    do
    {
        printf("\nIngrese un valor para b (divisor) menor que %d y distinto de cero: ",numa);
        scanf("%d",&numb);
        if(numb>numa or numb==0)
        {
            printf("\nEl numero tiene que ser menor que %d y distinto de cero. Ingrese nuevamente\n",numa);
            system("pause");
        }
    }while(numb>numa or numb==0);
    resultado=division(numa,numb);
    printf("Resultado de la division de a con b= %d",resultado);
}
```