

Introducción a las Funciones

[Anterior](#) [Siguiete](#) [Contenido](#)

Contenido del Capítulo:

[Introducción](#)
[Definición de una función](#)
[Dónde se define una función](#)
[Vida de una variable](#)
[Ejercicios](#)

Introducción

Vamos a dar un paso más en la complejidad de nuestros programas. Vamos a empezar a usar funciones creadas por nosotros. Las funciones son de una gran utilidad en los programas. Nos ayudan a que sean más legibles y más cortos. Con ellos estructuramos mejor los programas.

Una función sirve para realizar tareas concretas y simplificar el programa. Nos sirve para evitar tener que escribir el mismo código varias veces. Ya hemos visto en el curso algunas funciones como printf, gotoxy, scanf y clrscr. Estas funciones están definidas en una biblioteca (la biblioteca estándar de C). No tenemos que preocuparnos de ella porque el compilador se encarga de ella automáticamente.

Sin embargo nosotros también podemos definir nuestras propias funciones. Pocas veces se ve un programa un poco complejo que no use funciones. Una de ellas, que usamos siempre, es la función main.

Definición de una función

Una función tiene el siguiente formato:

```
tipo_de_variable nombre_de_la_función( argumentos )
{
    definición de variables;

    cuerpo de la función;

    return 0;
}
```

Poco a poco, empecemos por el nombre_de_la_función. Para el nombre no se pueden usar mas que letras, números y el símbolo '_'. No se pueden usar ni acentos ni espacios. Además el nombre de la función debe empezar por una letra, no puede empezar con un número. El nombre de la función se usa para llamarla dentro del programa.

El tipo_de_variable: Cuando una función se ejecuta y termina debe devolver un valor. Este valor puede ser cualquiera de los tipos de variables que hemos visto en el capítulo de Tipos de datos (int, char, float, double) o un tipo de dato definido por nosotros (esto

lo veremos más tarde). El valor que devuelve la función suele ser el resultado de las operaciones que se realizan en la función, o si han tenido éxito o no.

También podemos usar el tipo void. Este nos permite que podamos devolver cualquier tipo de variable o ninguna.

Definición de variables: Dentro de la función podemos definir variables que sólo tendrán validez dentro de la propia función. Si declaramos una variable en una función no podemos usarla en otra.

Cuerpo de la función: Aquí es donde va el código de la función.

Return: Antes hemos indicado que la función devuelve un valor. La sentencia return se usa para esto. El dato que se pone después de return es el dato que se devuelve. Puede ser una constante o una variable. Debe ser del mismo tipo que *tipo_de_variable*.

Argumentos: Estos son variables que se pasan como datos a una función. Deben ir separados por una coma. Cada variable debe ir con su tipo de variable. Las funciones deben definirse antes de ser llamadas. En los ejemplos a continuación se llama a la función desde main, así que tenemos que definir las antes que main. Esto lo veremos en el apartado siguiente.

Ejemplo 1. Función sin argumentos que no devuelve nada:

Este programa llama a la función prepara pantalla que borra la pantalla y muestra el mensaje "la pantalla está limpia". Por supuesto es de nula utilidad pero nos sirve para empezar.

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

void prepara_pantalla()          /* No se debe poner punto
y coma aquí */
{
    clrscr();
    printf( "La pantalla está limpia\n" );
    return;          /* No hace falta devolver ningún valor,
muchas gente ni siquiera pone este return */
}

void main()
{
    prepara_pantalla();    /* Llamamos a la función */
}
```

Ejemplo 2. Función con argumentos, no devuelve ningún valor:

En este ejemplo la función **compara** toma dos números, los compara y nos dice cual es mayor.

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

void compara( int a, int b )    /* Metemos los parámetros
a y b a la función */
```

```
{
    if ( a>b ) printf( "%i es mayor que %i\n" , a, b );
    else printf( "%i es mayor que %i\n", b, a );

    return;
}

void main()
{
    int num1, num2;

    printf( "Introduzca dos números: " );
    scanf( "%i %i", num1, num2 );

    compara( num1, num2 ); /* Llamamos a la función con sus
dos argumentos */
}
```

Ejemplo 3. Función con argumentos que devuelve un valor.

Este ejemplo es como el anterior pero devuelve como resultado el mayor de los dos números.

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

int compara( int a, int b )          /* Metemos los parámetros
a y b a la función */
{
    int mayor;                      /* Esta función define su propia
variable, esta variable sólo se puede usar aquí */

    if ( a>b )
        mayor = a;
    else mayor = b;

    return mayor;
}

void main()
{
    int num1, num2;
    int resultado;

    printf( "Introduzca dos números: " );
    scanf( "%i %i", num1, num2 );

    resultado = compara( num1, num2 ); /* Recogemos el
valor que devuelve la función en resultado */
    printf( "El mayor de los dos es %i\n", resultado );
}
```

En este ejemplo podíamos haber hecho también:

```
printf( "El mayor de los dos es %i\n", compara( num1,
num2 ) );
```

De esta forma nos ahorramos tener que definir la variable 'resultado'.

Dónde se definen las funciones

Las funciones deben definirse siempre antes de donde se usan. Lo habitual en un programa es:

Sección	Descripción
Includes	Aquí se indican qué ficheros externos se usan
Definiciones	Aquí se definen las constantes que se usan en el programa
Definición de variables	Aquí se definen las variables globales (las que se pueden usar en TODAS las funciones)
Definición de funciones	Aquí es donde se definen las funciones
Función main	Aquí se define la función main.

Esta es una forma muy habitual de estructurar un programa. Sin embargo esto no es algo rígido, no tiene por qué hacerse así, pero es recomendable.

Se puede hacer de otra forma, también aconsejable. Consiste en definir después de las variables las cabeceras de las funciones, sin escribir su código. Esto nos permite luego poner las funciones en cualquier orden. Ejemplos:

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

void compara( int a, int b );           /* Definimos la cabecera
de la función */

void main()
{
    int num1, num2;
    int resultado;

    printf( "Introduzca dos números: " );
    scanf( "%i %i", num1, num2 );

    resultado = compara( num1, num2 );
    printf( "El mayor de los dos es %i\n", resultado );
}

int compara( int a, int b )           /* Ahora podemos poner el
cuerpo de la función donde queramos. */
/* Incluso después de
donde la llamamos (main) */
{
    int mayor;
    if ( a>b )
        mayor = a;
    else mayor = b;

    return mayor;
}
```

Cuando se define la cabecera de la función sin su cuerpo (o código) debemos poner un ';' al final. Cuando definamos el cuerpo más tarde no debemos poner el ';', se hace como una función normal.

La definición debe ser igual cuando definimos sólo la cabecera y cuando definimos el cuerpo. Mismo nombre, mismo número y tipo de parámetros y mismo tipo de valor devuelto.

Vida de una variable

Cuando definimos una variable dentro de una función, esa variable sólo es válida dentro de la función. Si definimos una variable dentro de main sólo podremos usarla dentro de main. Si por el contrario la definimos como variable global, antes de las funciones, se puede usar en todas las funciones.

Podemos crear una variable global y en una función una variable local con el mismo nombre. Dentro de la función cuando llamamos a esa variable llamamos a la local, no a la global. Esto no da errores pero puede crear confusión al programar y al analizar el código.

Ejercicios

Ejercicio 1: Descubre los errores:

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

void main()
{
    int num1, num2;
    int resultado,

    printf( "Introduzca dos números: " );
    scanf( "%i %i", num1, num2 );
    resultado = compara( num1, num2 );
    printf( "El mayor de los dos es %i\n", resultado );
}

int compara( int a, int b );
{
    int mayor;
    if ( a>b )    mayor = a;
    else mayor = b;
    return mayor;
}
```

Solución:

- Hay una coma después de int resultado en vez de un punto y coma.
- Llamamos a la función compara dentro de main antes de definirla.
- Cuando definimos la función compara con su cuerpo hemos puesto un punto y coma al final, eso es un error.

Ejercicio 2: Busca los errores.

```
#include <stdio.h>

int resultado( int parametro )

void main()
```

```
{  
int a, b;  
  
a = 2; b = 3;  
printf( "%i", resultado( a ) );  
}  
  
char resultado( int parametro )  
{  
return parámetro+b;  
}
```

Solución:

- Hemos definido la cabecera de resultado sin punto y coma.
- Cuando definimos el cuerpo de resultado en su cabecera hemos puesto char, que no coincide con la definición que hemos hecho al principio.
- En la función resultado estamos usando la variable 'b' que está definida sólo en main. Por lo tanto es como si no existiera para resultado.
- En printf nos hemos dejado un paréntesis al final.

EJERCICIOS de C

Ejercicios Funciones Resueltos

Ejercicio: Realiza una función max que devuelva el número mayor de dos enteros.

```
#include<stdio.h>

int max(int x, int y);
{
    if (x < y)
        return y ;
    else
        return x ;
}

int main()
{
    int m, n;
    clrscr();

    do
    {
        scanf("%d %d", &m, &n);
        printf("Máximo de %d, %d es %d\n", max(m, n));    /*llama a max*/
    } while(m != 0);
    return 0;
}
```

Ejercicio: Calcular la media aritmética de dos números reales. Hazlo con funciones.

```
#include<stdio.h>

double media(double x1, double x2);
{
    return (x1+x2)/2 ;
}

int main()
{
    double num1, num2, med;
    clrscr();

    printf("Introducir dos números reales: ");
    scanf("%lf %lf", &num1, &num2);
}
```

```
        med=media(num1, num2);
        printf("El valor medio es %.4f \n", med);
return 0;
}
```

Ejercicio: Realiza una función que nos calcule el área del rectángulo, así como otra función para introducir los datos.

```
#include<stdio.h>

float area_rectangulo(float b, float a) ;
float entrada(void);

int main()
{
int b, h;
clrscr();

        printf("\n Base del rectángulo: ");
        b=entrada();
        printf("\n Altura del rectángulo: ");
        h=entrada();
        printf("\n Area del rectángulo:  %.2f", area_rectangulo(b,h));
return 0;
}

/* devuelve número positivo*/
float entrada()
{
float m;
do {
        scanf("%f", &m);
    } while (m<= 0.0);
return m;
}

/* calcula el área del rectángulo */
float area_rectangulo( float b, float a)
{
return (b*a);
}
```

Ejercicio: Función que nos calcule el cuadrado de un número.

```
#include<stdio.h>

double cuadrado(double n);

int main()
{
```



```
double x=11.5;
clrscr();

    printf("\n %6.2lf al cuadrado = %8.4lf \n", x, cuadrado(x));
return 0;
}

/* calcula el cuadrado */
double cuadrado (double n
{
    return (n*n);
}
```

Ejercicio: Función que nos visualiza un nombre.

```
#include<stdio.h>

void visualizar_nombre(char *nom) ;

void main()
{
clrscr();

    visualizar_nombre("Lucas El Fuerte");
}

/* Función visualizar */
void visualizar_nombre(char *nom)
{
    printf("Hola %s \n", nom);
}
```

Creación de macros con argumentos

Ejercicio: Aplicación completa de una macro con argumentos es Volcono(), que calcula el volumen de la figura geométrica Cono.

$$(V=1/3 \pi r^2h)$$

```
#include<stdio.h>

#define PI 3,141592
#define Volcono(radio, altura) ((PI*(radio*radio)*altura)/3.0)

int main()
{
float radio, altura, volumen;
clrscr();

    printf("\nIntroduzca radio del cono: ");
    scanf("%f", &radio);
```

```
printf("\nIntroduzca altura del cono: ");
scanf("%f", &altura);
volumen=Volcono(radio, altura);
printf("\n El volumen del cono es: %.2f", volumen);
return 0;
}
```

Ejercicio:

Una aplicación de una variable static en una función es la que permite obtener la serie de números de Fibonacci. El ejercicio que se plantea es el siguiente: dado un entero n, obtener los n primeros números de la serie de Fibonacci.

La secuencia de números de Fibonacci: 0, 1, 1, 2, 3, 5, 8, 13, ... se obtiene partiendo de los números 0, 1 y a partir de ellos cada número se obtiene sumando los dos anteriores:

$$a_n = a_{n-1} + a_{n-2}$$

La función fibonacci() que se escribe a continuación, tiene dos variables static, x e y. Se inicializan x a 0 e y a 1; a partir de esos valores se calcula el término actual, y, dejando preparado x para la siguiente llamada. Al ser variables static mantienen el valor entre llamada y llamada.

```
#include<stdio.h>

long int fibonacci(void);

int main()
{
int n, i;
clrscr();
printf("\nCuantos números de Fibonacci?: ");
scanf("%d",&n);
printf("\nSecuencia de fibonacci: 0, 1"

        for(i=2; i<n;i++)
            printf(", %ld", fibonacci());
return 0;
}
long int fibonacci(void)
{
static int x=0;
static int y=1;

        y=y+x;
        x=y-x;
return y;
}
```

Ejecución:

Cuantos números de Fibonacci ? : 14

Secuencia de fibonacci: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233

**Ejercicio: Programa que nos dice si una persona es mayor de edad.
Realízalo con funciones.**

```
#include<stdio.h>
int mayor(int d);
int main()
{
int edad;
clrscr();
printf("introduce una edad\n");
scanf("%d",&edad);
mayor(edad);
puts("fin de programa");
getch();
return 0;
}
int mayor(d)
{
if (d>0)
{
if(d>=18)
puts("VERDAD");
else
puts("FALSO");
}
else
printf("Valor erroneo\n");

return 0;
}
```

Ejercicio: Menú de funciones que realiza:

A.-suma de los números.

B.-raiz cuadrada de pares y de impares.

```
#include<stdio.h>
#include<ctype.h>
#include<math.h>
int sumanum();
float raizcuapar();
int main()
{
int S;
float R,R2;
char opcion;
clrscr();
printf("A.-suma de los numeros.\n");
printf("B.-raiz cuadrada de pares y de impares.\n");
printf("Introduzca una de las opciones:\n");
scanf("%c",&opcion);
opcion=toupper(opcion);
switch(opcion)
{
case 'A':
S=sumanum();
printf("El resultado de la suma es:%d",S);
break;
case 'B':
raizcuapar();
break;
}
}
int sumanum()
{
int i,sum=0;
for(i=1;i<=500;i+=8)
{
sum=sum+i;
}
return sum;
}
float raizcuapar()
{
int sumapares=0,sumaim=0,numero=1;
float rp,ri;
while(numero!=0)
{
numero=0;
printf("Introduzca un numero:\n");
scanf("%d",&numero);
```

```
    if(numero%2==0)
        sumapares=sumapares+numero;
    else
        sumaim=sumaim+numero;
}
rp=sqrt(sumapares);
ri=sqrt(sumaim);
printf("La raiz cuadrada de los pares es:%f\n",rp);
printf("La raiz cuadrada de los impares es:%f",ri);
return 0;
}
```

Ejercicio: Realiza un menú que realice la suma de varios número sabiendo desde que número hasta que número deseo y la siguiente opción no s realiza una función.

```
#include <stdio.h>
#include <math.h>
float suma(float n,float m);
float funcion(float x,float y);
int main()
{
    int opcion;
    float a=0,b=0,total,c=0,d=0;
    clrscr();
    printf("Menu\n");
    printf("1.-Suma de fracciones\n");
    printf("2.-Funcion\n");
    printf("Introduce una opcion del menu\n");
    scanf("%d",&opcion);
    switch(opcion)
    {
        case 1 : total=suma(c,d);
                printf("secuencia:%f\n",total);
                break;
        case 2 : total=funcion(a,b);
                printf("La funcion es: %f\n",total);
                break;
        default: printf("error\n");
    }
    printf("Fin de programa\n");
    getch();
    return 0;
}
float suma(float n,float m)
{
    float total,c,d,n=1,m=2;
    int cont;
    printf("Introduce dos numeros tope\n");
    scanf("%f %f",&c,&d);
```

```
while((n<c)&&(m<d));
{
total=n/m;
n=n+2;
m=m+2;
}
return total;
}
int funcion(float x,float y)
{
float total,c;
float x,y;
printf("Introduce un numero\n");
scanf("%f",&c);
printf("introduce un valor para a, y otro para b\n");
scanf("%f %f",&x,&y\n");
if(c>1)
{
total=x*y;
}
if(c=1);
{
total=x/y;
}
if(c<1);
{
total=pow(x,2)*pow(y,2);
}
return total;
}
```

Ejercicio: Realiza un programa que nos calcule la media y el área de dos números. Realiza con un menú y funciones.

```
#include<stdio.h>
float media(float m,float n);
float area(float m,float n);
int main()
{
int x;
float a,b;
float total;
clrscr();
printf("Menu\n");
printf("1.-Media\n");
printf("2.-Area\n");
printf("introduce una opcion del menu\n");
scanf("%d",&x);
printf("Introduce dos numeros para hacer las operaciones\n");
scanf("%f %f",&a,&b);
switch(x)
{
case 1: total=media(a,b);
printf("la media es:%.2f\n",total);
break;
case 2: total=area(a,b);
printf("el area es%.2f\n",total);
break;
default:
printf("no existe esa opcion\n");
}
printf("fin de programa");
getch();
return 0;
}
float media (float m,float n)
{
float total;
total=(m+n)/2;
return total;
}
float area(float m,float n)
{
float total;
total=m*n;
return total;
}
```

Ejercicio: Realiza con funciones un programa que nos haga un sumatoria de n números y también por otro lado una función que si el número pasado es mayor de 0 hago y^2+y y si no y^3+b .

```
/*programa menu*/
#include <stdio.h>
#include <math.h>
#include <ctype.h>
int sumatoria(int a);
int funcion(int b);
int main()
{
int n,suma1,funcion1;
char opcion;
clrscr();
printf("menu\n");
printf("A.-sumatoria\n");
printf("B.-calcula la funcion\n");
printf("elige una opcion del menu\n");
scanf("%c",&opcion);
printf("introduce un numero\n");
scanf("%d",&n);
opcion=toupper(opcion);
switch(opcion)
{
case 'A':
suma1=sumatoria(n);
printf("la suma total es:%d\n",suma1);
break;
case 'B': funcion1=funcion(n);
printf("el total de la funcion es:%d\n",funcion1);
break;
default: printf("error\n");
break;
}
getch();
return 0;
}
int sumatoria(int a)
{
int cont,b,suma1=0;
printf("introduce un valor tope\n");
scanf("%d",&b);
for(cont=0;cont<=b;cont++)
{
suma1=suma1+(a*cont);
}
return suma1;
}
int funcion(int b)
```



```
{  
int funcion1=0;  
if(b<0)  
funcion1=pow(b,2)+b;  
else  
funcion1=b+pow(b,3);  
return funcion1;  
}
```

Ejercicio: Diseña un programa con funciones para el cálculo del factorial de cualquier número deseado.

```
/*programa factorialfuncion*/  
#include<stdio.h>  
int factorial(int a);  
int main()  
{  
int n,total;  
clrscr();  
printf("introduce un numero positivo para hacer el factorial\n");  
scanf("%d",&n);  
total=factorial(n);  
printf("El factorial es:%d",total);  
getch();  
return 0;  
}  
int factorial(int a)  
{  
int n,total=1;  
while(a != 0)  
{  
total=total * a;  
a=a -1;  
}  
return total;  
}
```

Ejercicio: Programa con menú que realiza la suma, resta, multiplicación división y módulo de dos números. Hazlo con funciones.

```
#include <stdio.h>  
int suma(int m, int n);  
int rest(int m, int n);  
int mult(int m, int n);  
int divi(int m, int n);  
int mod(int m, int n);  
int main()  
{  
int a,b,total;  
int numero;
```



```
{
int total;
total= (m+n);
return total;
}
int rest(int m, int n)
{
int total;
total= (m-n);
return total;
}
int mult(int m, int n)
{
int total;
total= (m*n);
return total;
}
int divi(int m, int n)
{
int total;
total= (m/n);
return total;
}
int mod(int m, int n)
{
int total;
total= (m%n);
return total;
}
```

Ejercicio: Realiza un programa que nos haga el promedio de los números pares, de los impares y los múltiplos de 3 de n números. Hazlo con funciones.

```
/*programa menu tercero*/
#include <stdio.h>
#include <ctype.h>
float promediop(int n);
float promedioi(int n);
float multiplo(int m);
int main()
{
int a,b,suma,media,mediai,mediap;
char opcion;
clrscr();
printf("Menu\n");
printf("a.-Promedio pares e impares\n");
printf("b.-Multiplo de 3\n");
printf("Eliga una de las opciones\n");
scanf("%c",&opcion);
```

```
opcion=toupper(opcion);
switch(opcion)
{
case 'b': printf("introduce el total\n");
          scanf("%d",&a);
          suma=multiplo(a);
          printf("El resultado es:%d\n",suma);
          break;
case 'a': printf("introduce el total\n");
          scanf("%d",&a);
          mediap=promediop(a);
          mediai=promedioi(a);
          printf("la media de los pares es:%d\n",mediap);
          printf("la media de los impares es:%d\n",mediai);
          break;
default:
          printf("Error\n");
          break;
}
getch();
return 0;
}
float multiplo(int m)
{
int suma=0,i;
for(i=0;i<=m;i++)
{
if(i%3==0)
suma=suma+i;
}
return suma;
}
float promediop(int n)
{
int mediap=0,cont;
for(cont=0;cont<=n;cont++)
{
if(n%2==0)
mediap=mediap+n;
}
return mediap;
}
float promedioi(int n)
{
int mediai=0,cont;
for(cont=0;cont<n;cont++)
{
if(n%2!=0)
mediai=mediai+n;
}
}
```

```
return mediai;  
}
```

Ejercicio: Elabora un programa que nos realice la suma de los 500 primeros números. También deseo que nos de la raíz cuadrada de los números pares e impares por separado, parándose cuando introduzcamos el 0. utilice el switch y funciones.

```
#include<stdio.h>  
#include<math.h>  
int suma();  
float raizp();  
float raizi();  
int main()  
{  
int n,total,a,b;  
clrscr();  
printf("Menu\n");  
printf("1.-Suma de numeros\n");  
printf("2.-Raiz cuadrada\n");  
printf("introduce una opcion\n");  
scanf("%d",&n);  
switch(n)  
{  
case 1 : total=suma();  
printf("Suma de numeros entre 1 y 500 es:%d\n",total);  
break;  
case 2 : b=raizp();  
printf("Raiz cuadrada de n pares:%d\n",b);  
a=raizi();  
printf("Raiz cuadrad de n impares:%d\n",a);  
break;  
default: printf("error\n");  
break;  
}  
getch();  
return 0;  
}  
int suma()  
{  
int total=0,n;  
for(n=0;n<=500;n+=8)  
{  
total = total+n;  
}  
return total;  
}  
float raizp()  
{
```

```
int sumap=0,cont=0,n=1;
float b;

while(n!=0)
{
n=0;
printf("inserte un numero");
scanf("%d",&n);
if(n%2==0)

    sumap=sumap+n;
else
    break;

cont++;
}
    b=sqrt(sumap);
return b;
}
float raizi()
{
int sumai=0,cont=0,n=1;
float a;

while(n!=0)
{
n=0;
printf("inserte un numero");
scanf("%d",&n);
if(n%2!=0)
    sumai=sumai+n;
else
    break;

cont++;
}
    a=sqrt(sumai);
return a;
}
```

Ejercicio: Realiza un programa que nos realice:

1.-Sumatorio de los n primeros multiplos de 3.

2.-Imprimir el promedio de los n numeros pares e impares.

```
/*programa menu tercero*/
#include <stdio.h>
int sumatorio(int a);
int promedio(int b);
int main()
{
```

```
int n,total,cont,opcion,media=0,suma=0,mediapar,mediaimpar;
clrscr();
printf("Menu\n");
printf("1.-Sumatorio de los n primeros multiples de 3\n");
printf("2.-Imprimir el promedio de los n numeros pares e impares\n");
printf("Introduce una opcion del menu\n");
scanf("%d",&opcion);
printf("introduce el total de numeros\n");
scanf("%d",&total);
for(cont=1;cont<=total;cont++)
{
printf("introduce numero\n");
scanf("%d",&n);
}
switch(opcion)
{
case 1: suma=sumatorio (n);
printf("total:%d\n",suma);
break;
case 2: media=promedio (n);
printf("total:%d\n",mediapar);
printf("total:%d\n",mediaimpar);
break;
default:
printf("error\n");
}
printf("fin de programa\n");
getch();
return 0;
}
int sumatorio(int a)
{
int total,suma=0;

if(a%3==0);
{
suma=suma+a;
}
return suma;
}
int promedio(int b)
{
int media,mediapar=0,mediaimpar=0;
if(b%2==0);
{
mediapar=mediapar+b;
}
if(b%2!=0);
{
mediaimpar=mediaimpar+b;
}
```

```
}  
return mediaimpar;  
}
```

Ejercicio: Programa que nos realice:

- A- Media de las notas**
- B- Media de los aprobados**
- C- Media de los suspensos**
- D- Raiz Cuadrada**

```
#include<stdio.h>  
#include<ctype.h> /*para introducir las mayusculas*/  
#include<math.h> /*para hacer la operacion de raiz cuadrada*/  
float medianotas(int a);  
float mediasusp(int b);  
float mediaaprob(int c);  
float raiz(int x);  
int main()  
{  
float mnotas,msusp,maprob,raiz1;  
int n;  
char opcion;  
clrscr();  
printf("A- Media de las notas\n");  
printf("B- Media de los aprobados\n");  
printf("C- Media de los suspensos\n");  
printf("D- Raiz Cuadrada\n");  
printf("Elige una de las opciones:\n");  
scanf("%c",&opcion); /*c se utiliza pq es un caracter*/  
opcion=toupper(opcion);  
printf("Cuantos alumnos hay?\n");  
scanf("%d",&n);  
    switch(opcion)  
    {  
case 'A': /*comillas pq es un caracter*/  
    mnotas=medianotas(n);  
    printf("La media de las notas es:%f",mnotas);  
    break;  
case 'B':  
    msusp=mediasusp(n);  
    printf("La media de los suspensos es:%f",msusp);  
    break;  
case 'C':  
    maprob=mediaaprob(n);  
    printf("La media de los aprobados es:%f",maprob);  
    break;  
case 'D':  
    raiz1=raiz(n);  
    printf("La raiz de la suma total es:%.2f",raiz1);
```



```
        break;
default:
    printf("Error\n");
    break;
}
return 0;
}
float medianotas(int a)
{
    float suman=0,mnotas,notas;
    int i;
    for(i=0;i<a;i++)
    {
        printf("Introduzca la nota el alumno:\n");
        scanf("%f",&notas);
        suman=suman+notas;
    }
    mnotas=suman/a;
    return mnotas;
}
float mediaaprob(int c)
{
    float sumaaprob=0,maprob,nota;
    int i,cont=0;
    for(i=0;i<c;i++)
    {
        printf("introduzca la nota del alumno\n");
        scanf("%f",&nota);
        if(nota>=5);
        sumaaprob=sumaaprob+nota;
        cont=cont+1;
    }
    maprob=sumaaprob/cont;
    return maprob;
}
float mediasusp(int b)
{
    float sumasusp=0,msusp,nota;
    int i,cont=0;
    for(i=0;i<b;i++)
    {
        printf("introduzca la nota del alumno\n");
        scanf("%f",&nota);
        if(nota<5);
        sumasusp=sumasusp+nota;
        cont=cont+1;
    }
    msusp=sumasusp/cont;
    return msusp;
}
```

```
float raiz(int x)
{
    float raiz1, notas, suma=0;
    int cont;
    for(cont=0; cont<x; cont++)
    {
        printf("Introduzca la nota del alumno\n");
        scanf("%f", &notas);
        suma=suma + notas;
    }
    raiz1=sqrt(suma);
    return raiz1;
}
```

Ejercicio: Realiza un programa en el cual se obtenga:

A-Suma de fracciones

B-Funciones

```
#include <stdio.h>
#include <math.h>
#include <ctype.h>
float fracciones(int n);
float funcion();
int main()
{
    char opcion;
    int m;
    float frac, fun;
    clrscr();

    printf("Menu:\n");
    printf("A-Suma de fracciones\n");
    printf("B-Funciones\n");
    printf("Introduzca una de las opciones:\n");
    scanf("%c", &opcion);
    opcion=toupper(opcion);

    switch(opcion)
    {

    case 'A':    printf("Introduce hasta que numero de la serie:");
                scanf("%d", &m);
                frac=fracciones(m);
                printf("El total de las sumas de las fracciones es: %.2f\n", frac);
                break;

    case 'B':
                fun=funcion();
                printf("El valor de la funcion es: %f\n", fun);
    }
```

```
        break;

default:    printf("Opcion no valida");
           break;

}
getch();
return 0;
}

float fracciones(int n)
{
int i=0,j;
float total=0,sumap=0,sumai=0;
int a=2,b=1;
while(i<n)
{

    sumap=sumap+a;
    sumai=sumai+b;
    total=total+(sumai/sumap);
    a+=2;
    b+=2;
    i++;
}
return total;
}

float funcion ()
{
int total,c,a,b;
printf("Introduzca el parametro c:\n");
scanf("%d",&c);
printf("Introduzca valores para a y b:\n");
scanf("%d%d",&a,&b);

if(c>1)
{
total=(a*b);
}
if(c==1)
{
total=(a/b);
}
if(c<1)
{
total=pow(a,2)*pow(b,2);
}
return total;
}
```

}

Ejercicio: Programa que hace:

- 1.-Potencia de dos mayor que mil
- 2.-Edad

```
/*programa menu tercero*/
#include <stdio.h>
#include<math.h>
int potencia(int n);
int edad(int n);
int main()
{
int opcion,ed,x;
clrscr();
printf("Menu\n");
printf("1.-Potencia de dos mayor que mil\n");
printf("2.-Edad\n");
printf("Eliga una de las opciones\n");
scanf("%d",&opcion);
switch(opcion)
{
case 1 : potencia(x);
        break;
case 2 : printf("introduce una edad\n");
        scanf("%d",&ed);
        edad(ed);
        break;

default:
        printf("Error\n");
        break;
}
getch();
return 0;
}
int potencia(int n)
{
int x=1024;
printf("%d",x);
return x;
}
int edad(int n)
{
int ed;
if((n>=0)&&(n<=10))
printf("ninio\n");
if((n>=11)&&(n<=15))
printf("adolescente\n");
```

```
if((n>=16)&&(n<=25))
printf("pavo\n");
if((n>=26)&&(n<=40))
printf("Jovenes\n");
if((n>=41)&&(n<=65))
printf("adulto\n");
if(n>65)
printf("Viejo\n");
return ed;
}
```

Programa que nos indica si un año es bisiesto o no.

```
/*programa bisiestofuncion*/
#include <stdio.h>
int bisiesto(int f);
int main()
{
int a;
clrscr ();
printf("introduce un anio\n");
scanf("%d",&a);
bisiesto(a);
printf("fin de programa\n");
getch();
return 0;
}
int bisiesto(a)
{
if(a>0)
{
if ((a%4==0) && (a%100!=0) || (a%400==0))
printf("el anio es bisiesto\n");
else
printf("el anio no es bisiesto\n");
}
else
printf("el año introducido no es correcto\n");
return 0;
}
```

Ejercicio: Función que lee calificaciones (0-10) de alumnos y cuente el número de alumnos reprobados, (calif<7) de un conjunto de notas. La función recibe como parámetro el número de notas a leer y deberá regresar el número de alumnos reprobados. Crear un programa que llame a la función anterior y que despliegue el número de alumnos reprobados.

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

```
int numreprob(int not);
```

```
void main(void){  
int notas,reprobados;  
clrscr();  
  
printf("\nNumero de notas a leer: ");  
scanf("%d",&notas);  
reprobados=numreprob(notas);  
printf("\nExiste un total de %d alumnos reprobados",reprobados);  
getch();  
}
```

```
int numreprob(int not){  
  
int c=1,calif,rep=0;  
  
while(c<=not){  
  
printf("\nCalificacion de la nota %d: ",c);  
scanf("%d",&calif);  
  
if(calif<7){  
rep++;}  
  
c++;  
}  
return(rep);  
}
```

Ejercicio: Función que lee las matrículas (números enteros) y las calificaciones (tipo char) de los alumnos de un grupo y las almacene en dos arreglos unidimensionales. La función deberá leer los datos hasta que se de una matrícula cero la cual indica el fin de captura. La función regresa el número de alumnos por el mecanismo de return. Después debe de imprimir el contenido de los arreglos.

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

```
void despliega(int *mats,char *califs,int numcap);  
int leedato(int *mats,char *califs,int maximo);
```

```
void main(void){  
int max=30,matr[29],numalumn;
```

```
char calif[29];

clrscr();

numalumn=leedato(matr,calif,max);
printf("\nEl numero de alumnos capturados es %d",numalumn);
despliega(matr,calif,numalumn);

}
int leedato(int *mats,char *califs,int maximo){

int n=0;

printf("\n *** Introduce la matricula y su calificacion. ***");
printf("\n *** Para terminar dale un cero a la matricula. ***");

while(1){

if(n>=maximo){
printf("\nEl arreglo se encuentra lleno");
getch();
break;
}
printf("\n\nMatricula del alumno %d: ",n+1);
scanf("%d",mats);

if(*mats==0){
break;
}
printf("\nCalificacion del alumno %d: ",n+1);
scanf("%d",califs);

mats++;
califs++;
n++;
}
return n;
}

void despliega(int *mats,char *califs,int numcap){

int i=1;
```

```
while(1){  
  
    if(i>numcap){  
        break;  
    }  
    printf("\nLa calificacion del alumno %d es %d",*mats,*califs);  
    i++;  
    mats++;  
    califs++;  
}  
    getch();  
}
```

Ejercicio: Programa que despliega una tabla de conversión de °C a °F y viceversa. El programa debe tener una función que pregunte el tipo de conversión deseada. El usuario tecleará "c" o "C" si la conversión va a ser de °C a °F y "f" o "F" si la conversión es de °F a °C, regresando el caracter leído por medio de return. Otra función preguntará el rango de valores de la tabla y el incremento, recibiendo como parámetros las direcciones de las variables donde se almacenarán esos datos. Otras dos funciones serán para desplegar cada tipo de tabla de conversión.

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

```
void despliegaFC(int *ran1,int *ran2,int *inc);  
void despliegaCF(int *ran1,int *ran2,int *inc);  
void leeDatos(int *ran1,int *ran2,int *inc);  
char tipoConv(void);
```

```
void main(void){
```

```
    int rango1,rango2,incremento;  
    char tipo;  
    clrscr();  
    tipo=tipoConv();  
    leeDatos(&rango1,&rango2,&incremento);
```

```
    if(tipo=='C'){  
        despliegaCF(&rango1,&rango2,&incremento);  
    }  
    else{  
        despliegaFC(&rango1,&rango2,&incremento);
```



```
    }  
}  
char tipoConv(void){  
    char caract;  
  
    printf("    TABLA DE CONVERSION DE GRADOS CENTIGRADOS  
A GRADOS FARENHEIT");  
    printf("\n        Y DE GRADOS FARENHEIT A GRADOS  
CENTIGRADOS");  
    printf("\n\nPara convertir de Centigrados a Farenheit oprime 'C'");  
    printf("\n\npara convertir de Farenheit a Centigrados oprime 'F'");  
  
    printf("\n\nTipo de Conversion (C o F): ");  
    caract=toupper(getch());  
  
    return caract;  
}  
  
void leeDatos(int *ran1,int *ran2,int *inc){  
  
    printf("\n\nValor inicial de la tabla: ");  
    scanf("%d",ran1);  
    printf("\n\nValor final de la tabla: ");  
    scanf("%d",ran2);  
  
    printf("\n\nIncremento de los valores de la tabla: ");  
    scanf("%d",inc);  
}  
  
void despliegaCF(int *ran1,int *ran2,int *inc){  
  
    int suma=*ran1,incremento=*inc,rango2=*ran2;  
    float conversion;  
  
    while(suma<=rango2){  
  
        conversion=(((suma)*9)/5)+32;  
        printf("\n%d §C ----- %.1f  
§F",suma,conversion);  
  
        suma=suma+incremento;  
    }  
    getch();  
}
```

```
}  
  
void despliegaFC(int *ran1,int *ran2,int *inc){  
  
    int suma=*ran1,incremento=*inc,rango2=*ran2;  
    int conversion;  
  
    while(suma<=rango2){  
  
        conversion=((suma-32)*5)/9;  
        printf("\n%d §F ----- %d  
§C",suma,conversion);  
  
        suma=suma+incremento;  
    }  
    getch();  
}
```