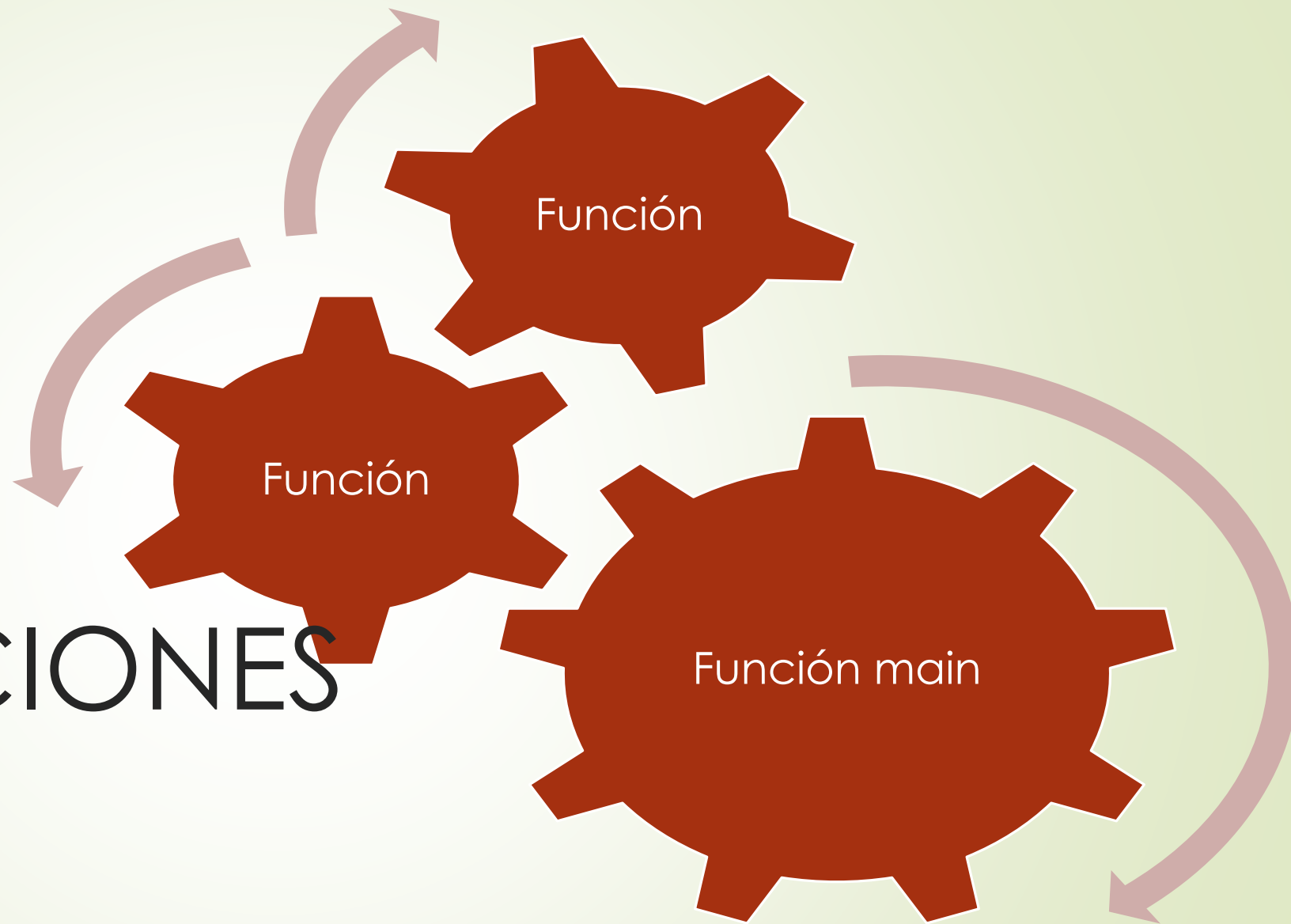


FUNCIONES



QUE ES UNA FUNCION?

SON PEQUEÑOS TROZOS DE PROGRAMAS O SUBROUTINAS, todo programa en C consta, básicamente de un conjunto de funciones, y una función llamada **MAIN**, la cual es la primera que se ejecuta al comenzar el programa, llamando desde ella al resto de las funciones que formen nuestro programa.

Las funciones nos sirven para dividir nuestro programa, en programitas pequeños. Ejemplo: cuando cocinamos un pasticho!!

El pasticho tiene varias etapas, que vamos realizando por separado, la salsa bechamel, el guiso de la carne, cocinamos la pasta(en caso de ser macarrones), estas pequeñas actividades son las funciones de nuestro programa,

Luego armamos el pasticho colocamos en un molde, el guiso de la carne, la pasta, el queso, jamón, salsa bechamel, y lo llevamos al horno para que este listo. (REPRESENTARIA LA FUNCION MAIN), en la cual llamamos a todas las funciones y agregamos las adicionales funciones preestablecidas o librerías (queso, jamón).



Pasti cho

Salsa
Becha
mel

Adicio
nales

Pasta

Guiso
Carne

```
#include <stdio.h>
int valor1,valor2;
int mayor (int x, int y);
int menor (int x, int y);
int main()
{int ma;
    printf("\n Ingrese el valor 1\t");
    scanf("%i",&valor1);
    printf("\n Ingrese el valor 2\t");
    scanf("%i",&valor2);
    // llamamos a la funcion
    ma=mayor(valor1,valor2);
    printf("\n\t el valor mayor es : %i",ma);

    printf("\n\t el valor mayor es :
%i",menor(valor1,valor2));
    return 0;
}
```

```
int mayor (int x, int y)
{return(x>y?x:y);
}
```

```
int menor (int x, int y)
{return(x<y?x:y);
}
```



El formato general de un función es:

```
TipoDato nombre(lista de Argumentos)
{ cuerpo de la function
  return (Dato);
}
```

return

- La sentencia RETURN, permite en primer lugar, salir de la función desde cualquier punto de ella, y en Segundo lugar devolver un valor del tipo de la función, si ello es necesario (no se devuelve ningún valor si la función es VOID).
- Ejemplo:

```
int mayor (int x, int y)
{
    if (x>y)
        return x;
    else
        return y;
}
```

- Observa que una función puede contener varios return, permite la posibilidad de salir de la función desde de distintos puntos, también termina su ejecución. Si llega al final sin encontrar una sentencia RETURN, quiere decir que es de tipo **VOID**.
- Ejemplo:

```
void mayor (int x, int y)
{
    if (x>y)
        printf("x es Mayor", x);
    else
        printf("y es Mayor", y);
}
```

Argumentos de las funciones

- ▀ Los argumentos son los datos necesarios para que nuestra función trabaje.
- ▀ Veamos el ejemplo anterior

```
int mayor (int x, int y)
{ if (x>y)
    return x;
  else
    return y;
}
```

Los valores (**int x, int y**), son los parámetros de la función mayor, los mismos al declarar la función se debe colocar el tipo y el nombre, tal y como declaramos una variable, y al colocar mas de uno separarlos por comas(.). En el ejemplo los parámetros **X** y **Y** son de tipo entero

También debemos notar que en el ejemplo anterior la función devuelve un valor **INT**, por lo cual en la sentencia **RETURN**, debe ir un valor d tipo entero, de lo contrario nos dará error.

Para utilizar nuestra función en el MAIN, solo debemos llamarla por su nombre y pasarle los parámetros en el mismo orden qu fueron declarados. En el ejemplo anterior, llamaríamos ala función asi;

```
ma=mayor(valor1,valor2);
```

Dond ma es una variable de tipo **INT**, que almacena el valor de retorno de la función **MAYOR**, el cual sus parámetros son las variables de tipo **INT** **VALOR1** y **VALOR2**, las cuales tomaran los **X=VALOR1** y **Y=VALOR2**.

Funciones que NO devuelven datos, y NO necesitan parametros

Una función puede NO devolver datos, así como puede NO tener parámetros, la misma puede solo ejecutarse y mostrar X información, ejemplo:

```
void menu()  
{   printf ("\n\n\t Menú de opciones ");  
    printf ("\n\t 1.- Indicar cual numero es mayor ");  
    printf ("\n\t 2.- Indicar cual numero es menor ");  
    printf ("\n\t 3.- Salir");  
    printf ("\n\n\t Seleccione una opción");  
}
```

La función **MENU**, no devuelve ningún dato, por lo cual no tiene la sentencia **RETURN**, y tiene **VOID** (vacío) al principio de su declaración.

De igual manera NO tiene parámetros, por lo tanto los paréntesis quedan vacío ().

NOTA: en los argumentos se puede colocar VOID, pero no es estrictamente necesario, es decir la declaración quedaría así,

void menu(void)

Funciones que NO devuelven datos, y necesitan parametros

Una función puede NO devolver datos, y necesitar parámetros. Ejemplo:

```
void par(int numero)
{ if ((numero%2)==0)
    {printf("\n\n Numero es Par");
    }
else
    {printf("\n\n Numero es Impar");
    }
}
```

La función **PAR**, no devuelve ningún dato, por lo cual no tiene la sentencia **RETURN**, y tiene **VOID** (vacío) al principio de su declaración.

De igual manera **SI** tiene parámetros, por lo cual necesita un valor entero (**int numero**) para ser ejecutada.

Funciones que **SI** devuelven datos, y **NO** necesitan parametros

Una función puede devolver datos, y NO necesitar parámetros. Ejemplo:

```
int validar()  
{if ((opc>4) || (opc<1))  
    {  
        //clrscr();  
        return(-1);  
    }  
    return (0);  
}
```

La función **VALIDAR**, devuelve un dato entero, devuelve -1 si es incorrecta y 0 si es correcta, por lo cual tiene la sentencia **RETURN**.

De igual manera **NO** tiene parámetros, por lo cual no necesita.

Funciones que **SI** devuelven datos, y **SI** necesitan parámetros

Una función puede devolver datos, y necesitar parámetros. Ejemplo:

```
int mayor (int x, int y)
{
    return(x>y?x:y);
}
```

La función **MAYOR**, devuelve un dato entero,, por lo cual tiene la sentencia **RETURN**.

De igual manera **SI** tiene parámetros, veras que puede tener no solo un parámetro sino multiplex, en este caso dos parámetros, lo que se debe tener cuidado al momento de pasar los datos a la función, debido que hay que pasarlo en el mismo orden que fueron declarados n la función.

Acotaciones

- Las funciones deben ser declaradas al principio del programa, en el área de declaraciones, debajo de las llamadas a la librería, y de las declaraciones de las variables globales. (recuerda las variables globales están disponibles para todo el programa, en cambio las variables locales solo puedes acceder ELLAS EN el bloque o función donde fueron creadas)
- Las funciones pueden ser desarrolladas después del main, es decir abrimos llaves del main, lo cerramos y allí desarrollamos las mismas.
- Las funciones se deben llamar por el nombre indicado tal y como la defines, como se realiza con las variables.
- Veamos el siguiente ejemplo.

Ejemplo:

```
#include <stdio.h>
#include <ctype.h>
#include <stdlib.h>
int opc;
int valor1,valor2,n;// variables globales
int mayor (int x, int y);// funcion que indica cual es mayor
int menor (int x, int y);// funcion que indica cual es menor
void menu();// funcion muestra el menu
void cargaDatos();//carga los datos
void par(int numero);
int validar();
int main()
{int ma;//variable LOCAL
do{ menu();// llamado a funcion menu
    scanf("%d",&opc);
    switch(opc)
    { case 1: cargaDatos();// cargamos los datos
        ma=mayor(valor1,valor2); // llamamos a la funcion
        printf("\n\t el valor mayor es : %i",ma);// mostramos
        break;
    case 2:
        cargaDatos();// cargamos los datos
        // llamamos a la funcion y mostramos
        printf("\n\t el valor mayor es : %i",menor(valor1,valor2));
        break;
    case 3: printf("\n\n\n ingrese el numero");
        scanf("%d",&n);
        par(n);// llamamos a la funcion
        break;
```

```
case 4: printf("\n\n\n Hasta luego");
        break;
    default: if (validar())
        { system("cls");
          printf("\n\n\n ERROR OPCION INCORRECTA");
          printf("\n\n\n Debe ingresar un opcion del menu");
        }
    } // FIN SWITCH
}while(opc!=4);
return 0;
}

int validar()
{if ((opc>4) || (opc<1))
{
    //clrscr();
    return(-1);
}
return (0);
}

void par(int numero)
{ if ((numero%2)==0)
    {printf("\n\n Numero es Par");
    }
    else
    {printf("\n\n Numero es Impar");
    }
}

int mayor (int x, int y)
{return(x>y?x:y);
}

int menor (int x, int y)
{ return(x<y?x:y);
}

void menu()
{ printf ("\n\n\t Menu de opciones ");
  printf ("\n\t 1.- Indicar cual numero es mayor ");
  printf ("\n\t 2.- Indicar cual numero es menor ");
  printf ("\n\t 3.- Indicar numero es Par o Impar ");
  printf ("\n\t 4.- Salir");
  printf ("\n\n\t Seleccione una opcion");
}

void cargaDatos()
{printf("\n Ingrese el valor 1\t");
scanf("%i",&valor1);
printf("\n Ingrese el valor 2\t");
scanf("%i",&valor2);
}
```

Menu de opciones

- 1.- Indicar cual numero es mayor
- 2.- Indicar cual numero es menor
- 3.- Indicar numero es Par o Impar
- 4.- Salir

Seleccione una opcion_

Menu de opciones

- 1.- Indicar cual numero es mayor
- 2.- Indicar cual numero es menor
- 3.- Indicar numero es Par o Impar
- 4.- Salir

Seleccione una opcion1

Ingrese el valor 1 23

Ingrese el valor 2 32

el valor mayor es : 32

Menu de opciones

- 1.- Indicar cual numero es mayor
- 2.- Indicar cual numero es menor
- 3.- Indicar numero es Par o Impar
- 4.- Salir

Seleccione una opcion_

el valor mayor es : 32

Menu de opciones

- 1.- Indicar cual numero es mayor
- 2.- Indicar cual numero es menor
- 3.- Indicar numero es Par o Impar
- 4.- Salir

Seleccione una opcion2

Ingrese el valor 1 43

Ingrese el valor 2 45

el valor mayor es : 43

Menu de opciones

- 1.- Indicar cual numero es mayor
- 2.- Indicar cual numero es menor
- 3.- Indicar numero es Par o Impar
- 4.- Salir

Seleccione una opcion

Seleccione una opcion3

ingrese el numero32

Numero es Par

Menu de opciones

- 1.- Indicar cual numero es mayor
- 2.- Indicar cual numero es menor
- 3.- Indicar numero es Par o Impar
- 4.- Salir

Seleccione una opcion_

ERROR OPCION INCORRECTA

Debe ingresar un opcion del menu

Menu de opciones

- 1.- Indicar cual numero es mayor
- 2.- Indicar cual numero es menor
- 3.- Indicar numero es Par o Impar
- 4.- Salir

Seleccione una opcion_

Hacer un programa que Encripte y devuelva una cadena de caracteres ingresada por el usuario. Para encriptar, cambiaremos 3 posiciones

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
char cadena[120];
int validar();
void cargardatos();
int encriptar(char cadena[]);
void devolver(char cadena[]);
void mostrar();
void menu();
int main()
{int opc;
do{
    menu();
    scanf("%d",&opc);
    fflush(stdin);
    switch(opc)
    {case 1: system("cls");
        cargardatos();
        break;
```

```
case 2: system("cls");
        if (encriptar(cadena)==0)
            printf("\n cadena Encriptada\n");
        else
            printf("\n Error al encriptar\n");
        break;
        case 3:    system("cls");
        devolver(cadena);
        break;
        case 4: system("cls");
        mostrar();
        break;
        case 5: printf("\n\n\n Hasta luego");
        break;
        default: if ((opc>5) || (opc<1))
        { system("cls");
            printf("\n\n\n ERROR OPCION INCORRECTA");
            printf("\n\n\n Debe ingresar un opcion del menu");}
        }// FIN SWITCH
}while(opc!=5);
return 0;}
```

```
void menu()
{printf ("\n\n\t Menu de opciones ");
printf ("\n\t 1.- Cargar Datos ");
printf ("\n\t 2.- Encriptar ");
printf ("\n\t 3.- Des-Encriptar ");
printf ("\n\t 4.- Mostrar");
printf ("\n\t 5.- Salir");
printf ("\n\n\t Seleccione una opcion\t");}
void cargardatos()
{puts("Ingrese la cadena de caracteres a encriptar");
gets(cadena);
printf("\n \n Datos Almacenados"); }
void mostrar()
{puts("\n");
    puts(cadena);}
int encriptar(char cadena[])
{ int i=0;//variable local
    for(i=0;i<(strlen(cadena));i++)
    { if (cadena[i]!='\0')
        cadena[i]=cadena[i]+3;}
    if (i>=strlen(cadena))
        return (0);
    else return (1);}
void devolver(char cadena[])
{int i;// variable local
    for(i=0;i<(strlen(cadena));i++)
    { if (cadena[i]!='\0')
        cadena[i]=cadena[i]-3;}}
```



```
funciones1.1.c - C:\Users\Ing.Ordoñez\Des
C:\Windows\SYSTEM32\cmd.exe

Menu de opciones
1.- Cargar Datos
2.- Encriptar
3.- Des-Encriptar
4.- Mostrar
5.- Salir

Seleccione una opcion
```

```
Ingrese la cadena de caracteres a encriptar
ola

Datos Almacenados

Menu de opciones
1.- Cargar Datos
2.- Encriptar
3.- Des-Encriptar
4.- Mostrar
5.- Salir

Seleccione una opcion _
```

```
C:\Windows\SYSTEM32\cmd.exe

krod

Menu de opciones
1.- Cargar Datos
2.- Encriptar
3.- Des-Encriptar
4.- Mostrar
5.- Salir

Seleccione una opcion 2_

C:\Windows\SYSTEM32\cmd.exe

ola

Menu de opciones
1.- Cargar Datos
2.- Encriptar
3.- Des-Encriptar
4.- Mostrar
5.- Salir

Seleccione una opcion 5

Hasta luego
-----
program exited with code: 0>
presione una tecla para continuar . . .
```



Ejercicio

- Hacer un programa aplicando funciones, en el cual se puedan pedir 5 notas de un alumno y muestre su promedio. Debe contener un menú de opciones.
- Hacer un programa aplicando funciones donde, se pueda indicar dado un numero por el usuario, si el numero es primo o no, es par o impar. Desglosar en funciones cada proceso y luego llamarlo en el main.

Evaluación 2do Corte

- El segundo corte era evaluado de la siguiente manera

10% en dos actividades

Actividad 1: Arreglos

Actividad 2: Arreglos Bidimensionales

15% 1 Actividad

Actividad 3 Funciones

Para la actividad de funciones se estará evaluando la aplicación de todos los conocimientos adquiridos hasta el momento.

Debido a problemas de conexión datos, estaré recibiendo todas las actividades pendientes hasta el 25 de abril.