Practica No.4

Unidad temática: *III Manejo de funciones y cadenas*

Objetivos:

- Crear funciones y hacer llamado de estas para resolver problemas.
- Reutilizar código y aislar mejor los problemas.
- Emplear la recursividad de funciones en la solución de problemas.

Introducción

¿Qué es una función?

El código de un programa escrito en C se divide en funciones. Una función en C se distingue ***sólo*** por su nombre.

Las funciones suelen encapsular una operación más o menos compleja de la que se deriva un resultado. Para ejecutar esta operación, las funciones pueden precisar la invocación de otras funciones (o incluso de ellas mismas como es el caso de las funciones recursivas).

Las funciones en un programa son entidades que dado un conjunto de datos (los parámetros), se les encarga realizar una tarea muy concreta y se espera hasta obtener el resultado. Lo idóneo es dividir tareas complejas en porciones más simples que se implementan como funciones. La división y agrupación de tareas en funciones es uno de los aspectos más importantes en el diseño de un programa.

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.

Desarrollo

Tipos de funciones que se puede tener en C, son:

DESCRIPCIÓN	Terminología de C
Algoritmo principal	Función main
Módulo sin parámetros de entrada	Función sin parámetros de entrada
Módulo genérico con parámetros de entrada	Función con parámetros de entrada
Módulo de código que se ejecuta cuando es llamado desde algún punto del programa y no devuelve un valor	Función con tipo de retorno nulo (void). También se dice que es una función sin tipo de retorno.
Módulo de código que se ejecuta cuando es llamado desde algún punto del programa y devuelve un valor	Función con un tipo de retorno

Una función devuelve un valor, de ahí que especifiquemos un tipo de dato para ella, que hemos indicado como tipoDeRetorno. En caso de que el tipo indicado en lugar de ser un tipo de dato válido en C (como int, double o cualquier otro) sea **void**, la función **no devolverá nada** y en lugar de terminar con return valorDevueltoPorLaFuncion; la terminaremos simplemente con return; En realidad return(No es completamente necesario).

El flujo para una función sigue las reglas ya conocidas: al llegar el control a la sentencia return el flujo del programa vuelve a la sentencia inmediatamente posterior a la llamada efectuada. Si existe código posterior a la sentencia return final, éste será ignorado.

Las funciones pueden insertarse en el programa en cualquier orden.

La llamada a una función se realiza, cuando no hay parámetros que pasar, simplemente escribiendo su nombre seguido de unos paréntesis vacíos. La llamada a una función se hará normalmente para obtener un valor o asignar un valor a una variable, en expresiones del tipo:

```
printf (nombreDeLaFunción());

variable = nombreDeLaFunción();

If (nombreDeLaFunción() > variable) …
```

Hay que recordar siempre que una "función" con tipo de retorno especificado ejecuta un código y devuelve un valor: podríamos decir que tiene una similitud importante con las variables: tener un valor.

Ejemplo de funciones con tipo de retorno void y las funciones con tipo de retorno especificado.

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

int sumaDosEnteros (int entero1, int entero2) {
   int resultado = 0;
   resultado = entero1 + entero2;
   return resultado;
}

int main() {
   printf("Bienvenidos al programa\n");
   printf("Si sumamos tres y cinco obtenemos %d\n",
sumaDosEnteros(3,5));
   return 0; // Ejemplos aprenderaprogramar.com
}
```

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

void sumaDosEnteros (int entero1, int entero2) {
   int resultado = 0;
   resultado = entero1 + entero2;
   printf("Si sumamos %d y %d obtenemos %d\n", entero1, entero2,
resultado);
   return; // Ejemplos aprenderaprogramar.com
}

int main() {
   printf("Bienvenidos al programa\n");
   sumaDosEnteros(3,5);
   return 0;
}
```

Paso de parámetros a una función

Utilizando la lista de argumentos podemos pasar parámetros a una función. En esta lista se suele colocar un conjunto de identificadores, separados por comas, que representan cada uno de ellos a uno de los parámetros de la función. Obsérvese que el orden de los parámetros es importante. Para llamar a la función habrá que colocar los parámetros en el orden en que la función los espera.

Cada parámetro puede tener un tipo diferente. Para declarar el tipo de los parámetros añadiremos entre el paréntesis ')' y la llave '{' una lista de declaraciones, similar a una lista de declaraciones de variables. Es habitual colocar cada par metro en una línea, tabulados hacia la derecha. Así:

```
Imprime (numero, letra)
int numero;
char letra;
{
        printf ("%d, %c\n", numero, letra);
}
```

es una función que admite dos variables, una entera y otra de tipo carácter.

En los lenguajes de programación estructurada hay dos formas de pasar variables a una función:

- **por referencia**, o
- **por valor**

Cuando la variable se pasa por referencia, la función puede acceder a la variable original. Este enfoque es habitual en lenguajes como el Pascal. En C, sin embargo, todos los parámetros se pasan por valor. La función recibe una copia de los parámetros y variables, y no puede acceder a las variables originales. Cualquier modificación que efectuemos sobre un parámetro no se refleja en la variable original. Esto hace que no podamos alterar el valor de la variable por equivocación.

CAPTURAS DE PANTALLA

Ejercicio 1

```c
#include <stdio.h> //Elaborado por RICARDO BALDERRABANO RODRIGUEZ
#include <stdlib.h>
//El programa calcula el volumen de 3 habitaciones, asi como su volumen total.
int vol_t=0;
int calcular_volumen(int alto,int ancho,int longitud){
      int vol=0;                //Funcion para calcular el volumen
      vol=alto*ancho*longitud;
      return vol;
      }
int volumen_total(int vol){  //Funcion para la suma de los vol.
      vol_t=vol_t+vol;
      return vol_t;
      }
int main(int argc, char *argv[])
{
int alto,ancho,longitud,vol,i,vol_t;
  printf("Este programa calculara el volumen de 3 habitaciones.\n");
  for(i=1;i<4;i++){
    printf("\nHabitacion %d\n",i);
    printf("Introduzca en metros\n");
    printf("Altura:");
    scanf("%d",&alto);
    printf("El ancho:");
    scanf("%d",&ancho);
    printf("La longitud:");
    scanf("%d",&longitud);
    vol=calcular_volumen(alto,ancho,longitud);
    vol_t=volumen_total(vol);
    printf("El volumen de la habitacion %d es:%d metros cubicos\n",i,vol);
}
printf("Volumen total: %d metros cubicos\n",vol_t);
system("PAUSE");
  return 0;
```

```
F:\UPIITA\Introduccion a la Programacion\Practica4\Ejer...
Este programa calculara el volumen de 3 habitaciones.

Habitacion 1
Introduzca en metros
Altura:2
El ancho:20
La longitud:10
El volumen de la habitacion 1 es:400 metros cubicos

Habitacion 2
Introduzca en metros
Altura:2
El ancho:15
La longitud:20
El volumen de la habitacion 2 es:600 metros cubicos

Habitacion 3
Introduzca en metros
Altura:2
El ancho:16
La longitud:30
El volumen de la habitacion 3 es:960 metros cubicos
Volumen total: 1960 metros cubicos
Presione una tecla para continuar . . .
```

Ejercicio 2

```c
//Programa elaborado por RICARDO BALDERRABANO RODRIGUEZ
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int res=0;
int main(int argc, char *argv[]) {
    int N,res;
    printf("Introduzca un numero entero:");
    scanf("%d",&N);          //Lee el numero
    res=calcular_serie(N);
    printf("El resultado de la serie es: %d\n",res);
    return 0;
}
int calcular_serie(int N){ //Funcion que calcula la serie
    if(N<1){
    printf("Introduzca un numero mayor a 0");
      }else if(N<2){
        return 1;
      } else {
        if(N%2==0){
            res=-pow(N,N)+(calcular_serie(N-1));
          }
          else{
           res=pow(N,N)+(calcular_serie(N-1));
          }
        }
       return res;
 }

```

```
F:\UPIITA\Introduccion a la Programacion\Practica4\Ejer...
Introduzca un numero entero:5
El resultado de la serie es: 2893

--------------------------------
Process exited after 2.719 seconds with return value 0
Presione una tecla para continuar . . .
```

Ejercicio 3

```c
//Codigo elaborado por RICARDO BALDERRABANO RODRIGUEZ
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]) {
    int base,exponente,res;  //Se ingresan los datos
    printf("Se clacula la potencia de un numero.\n");
    printf("Introduzca la base:");
    scanf("%d&d",&base);
    printf("Introduzca el exponente:");
    scanf("%d",&exponente);
    res=entero_potencia(base,exponente);
    printf("Resultado:%d",res);
    return 0;
}
int entero_potencia(int base,int exponente){
    int i,res=1;  //Funcion que calcula la potencia del numero
    for(i=0;i<exponente;i++){
     res=res*base;
     }
     return res;
    }

```

```
F:\UPIITA\Introduccion a la Programacic
Introduzca la base:2
Introduzca el exponente:5
Resultado:32
--------------------------------
Process exited after 7.352 sec

Presione una tecla para contin
```

Ejercicio 4

```c
//Codigo elaborado por RICARDO BALDERRABANO RODRIGUEZ
#include <stdio.h>
#include <stdlib.h>
float calcular_resistencia(float voltaje,float corriente){
    float resistencia=0;
    return voltaje/corriente;
}
float calcular_voltaje(float resistencia,float corriente){
    return resistencia*corriente;
}
float calcular_corriente(float voltaje,float resistencia){
    return voltaje/resistencia;
}
int main(int argc, char *argv[]) {
    float voltaje,resistencia,corriente;
    char opcion;
    printf("Se puede calcular la resistencia,voltaje y corriente.\n\n");
    printf("Resistencia----R\n");
    printf("Voltaje--------V\n");
    printf("Corriente------C\n");
while(1){
    printf("\nIntroduzca la letra de lo que se quiere calcular:");
    scanf("%s",&opcion);
    printf("Introduce el valor de:\n");
    switch (opcion){
    case 'R':
    printf("Voltaje:");
            scanf("%f",&voltaje);
            printf("Corriente");
            scanf("%f",&corriente);
            resistencia=calcular_resistencia(voltaje,corriente);
            printf("El valor de la resistencia es:%.2f",resistencia);
            break;
    case 'V': printf("Resistencia:");
            scanf("%f",&resistencia);
            printf("Corriente:");
            scanf("%f",&corriente);
            voltaje=calcular_voltaje(resistencia,corriente);
            printf("El valor del voltaje es:%.2f",voltaje);
            break;
    case 'C': printf("Voltaje:");
            scanf("%f",&voltaje);
            printf("Resistencia:");
            scanf("%f",&resistencia);
            corriente=calcular_corriente(voltaje,resistencia);
            printf("El valor de la corriente es:%.2f",corriente);
            break;
            default: printf("Caracter no valido.");
        }
    }
    return 0;
}
```

```
F:\UPIITA\Introduccion a la Programacion\Practica4\Ejercicios\Ejercicio 4\Ejercicio4_Practia4.exe
Se puede calcular la resistencia,voltaje y corriente.

Resistencia----R
Voltaje--------V
Corriente------C


Introduzca la letra de lo que se quiere calcular:R
Introduce el valor de:
Voltaje:5
Corriente.3
El valor de la resistencia es:16.67

Introduzca la letra de lo que se quiere calcular:V
Introduce el valor de:
Resistencia:50
Corriente:2
El valor del voltaje es:100.00

Introduzca la letra de lo que se quiere calcular:C
Introduce el valor de:
Voltaje:10
Resistencia:15
El valor de la corriente es:0.67
```

Ejercicio 5

```c
//Codigo elaborado por RICARDO BALDERRABANO RODRIGUEZ
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
double hipotenusa(double lado1,double lado2){
    return sqrt(pow(lado1,2)+pow(lado2,2)); //Funcion calcula
}                                           //la hipotenusa
int main(int argc, char *argv[]) {
    double lado1,lado2,res;   //Variables de los lados del triangulo
    int i=1;
    printf("Se calcula la hipotenusa de un triangulo recto.\n");
    while(1){
    printf("Introduzca dimensiones(m) del triangulo %d:\n",i);
    printf("El lado 1:");
    scanf("%lf",&lado1);
    printf("El lado 2:");
    scanf("%lf",&lado2);
    res=hipotenusa(lado1,lado2);  //Llamado a la funcion
    printf("El valor de la hipotenusa es:%.1f metros.\n\n",res);
    i=i+1;
    }
    return 0;
}
```

```
F:\UPIITA\Introduccion a la Programacion\Practica4\Ejercicio...
Se calcula la hipotenusa de un triangulo recto.
Introduzca dimensiones(m) del triangulo 1:
El lado 1:3
El lado 2:4
El valor de la hipotenusa es:5.0 metros.

Introduzca dimensiones(m) del triangulo 2:
El lado 1:5
El lado 2:12
El valor de la hipotenusa es:13.0 metros.

Introduzca dimensiones(m) del triangulo 3:
El lado 1:8
El lado 2:15
El valor de la hipotenusa es:17.0 metros.
```

Ejercicio 6

```c
//Codigo elaborado por RICARDO BALDERRABANO RODRIGUEZ
//Se calcula el factorial de un numero deseado.
#include <stdio.h>
#include <stdlib.h>
long factorial(long N){  //Funcion que calcula el factorial
    if(N<1) printf("Introduzca un numero mayor a 0.");
    else if(N<2) return 1;
    else{
        return N*factorial(N-1);
    }
}
int main(int argc, char *argv[]) {
    long N,res;
    printf("Se calcula el factorial de un numero.\n");
    printf("Introduzca el numero:");
    scanf("%ld",&N);
    res=factorial(N);  //Llamado a la funcion
    printf("El factorial es:%d",res);
    return 0;
}
```

```
F:\UPIITA\Introduccion a la Programacion\Practica4\Eje...
Se calcula el factorial de un numero.
Introduzca el numero:5
El factorial es:120
--------------------------------
Process exited after 8.639 seconds with ret
Presione una tecla para continuar . . .
```

Ejercicio de aplicación

```c
#include <stdio.h>
#include <stdlib.h>
#define pi 3.1416
#define g 9.81
#include <math.h>
float masa;
float carrera_cil=10; //Carrera max de 10 pulgadas
float area_efec(float diametro){ //Funcion para calcular el area
      return (pow(diametro,2)*pi)/4;
      }
float fuerza(float masa){ //Funcion para calcular el peso
    return masa*g;
}
float pre_hidra(float fuerza,float area_efec){ //Funcion para calcular la presion
      return fuerza/area_efec;                 //Hidraulica de cada carga
      }
float vol_cilindro(float fuerza,float carrera_cil, float pre_hidra){ //Funcion para
    return (fuerza*carrera_cil)/pre_hidra;                  //calcular el volumen
}
int main(int argc, char *argv[]) {
    float pe1,pe2,pe3,pe4,area,pre1,pre2,pre3,pre4,diametro,vol1,vol2,vol3,vol4;
    printf("Introduzca el diametro de cilindro(plg):");
    scanf("%f",&diametro); //Introducir el diametro de mi tubo
    area=area_efec(diametro); //Llamado de funcion
    printf("\nIntroduzca la masa.\n");
    printf("Carga A\n");
    printf("1:");
    scanf("%f",&masa);  //Introducir m1
    pe1=fuerza(masa);   //Llamado de funcion
    printf("2:",&masa);
    scanf("%f",&masa);  //Introducir m2
    pe2=fuerza(masa);   //Llamado de funcion
    printf("Carga B\n");
    printf("2:");
    scanf("%f",&masa);  //Introducir m3
    pe3=fuerza(masa);   //Llamado de funcion
    printf("3:",&masa);
    scanf("%f",&masa);   //Introducir m4
    pe4=fuerza(masa);   //Llamado de funcion
    printf("\nLos pesos de cada carga son:\n");
    printf("1:%.2f \n",pe1);  //Impresion de resultados
    printf("2:%.2f \n",pe2);
    printf("3:%.2f \n",pe3);
    printf("4:%.2f \n",pe4);
    printf("\nPresion hidraulica de trabajo\n");
    pre1=pre_hidra(pe1,area); //Llamado de funcion
    printf("1:%.2f psi\n",pre1);
    pre2=pre_hidra(pe2,area); //Llamado de funcion
    printf("2:%.2f psi\n",pre2);
    pre3=pre_hidra(pe3,area); //Llamado de funcion
    printf("3:%.2f psi\n",pre3);
    pre4=pre_hidra(pe4,area); //Llamado de funcion
    printf("4:%.2f psi\n",pre4);
    printf("\nVolumen del aceite del cilindro necesario:\n");
    vol1=vol_cilindro(pe1,carrera_cil,pre1); //Llamado de funcion
    printf("Vol para tuberia 1: %.2f inch cubicas\n",vol1);
    vol2=vol_cilindro(pe2,carrera_cil,pre2); //Llamado de funcion
    printf("Vol para tuberia 2: %.2f inch cubicas\n",vol2);
    vol3=vol_cilindro(pe3,carrera_cil,pre3); //Llamado de funcion
    printf("Vol para tuberia 3: %.2f inch cubicas\n",vol3);
    vol4=vol_cilindro(pe4,carrera_cil,pre4); //Llamado de funcion
    printf("Vol para tuberia 4: %.2f inch cubicas\n",vol4);
    system("PAUSE");
    return 0;
}
```

```
Introduzca el diametro de cilindro(plg):10

Introduzca la masa.
Carga A
1:5
2:10
Carga B
2:8
3:6

Los pesos de cada carga son:
1:49.05
2:98.10
3:78.48
4:58.86

Presion hidraulica de trabajo
1:0.62 psi
2:1.25 psi
3:1.00 psi
4:0.75 psi

Volumen del aceite del cilindro necesario:
Vol para tuberia 1: 785.40 inch cubicas
Vol para tuberia 2: 785.40 inch cubicas
Vol para tuberia 3: 785.40 inch cubicas
Vol para tuberia 4: 785.40 inch cubicas
Presione una tecla para continuar . . .
```

Bibliografía

http://www.it.uc3m.es/abel/as/DSP/M1/FunctionDef_es.html

http://programandoenc.over-blog.es/article-32481588.html

https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=947:funciones-en-c-ique-significa-void-ique-es-el-tipo-de-retorno-ipara-que-sirve-return-modulos-cu00547f&catid=82&Itemid=210