



Universidad Nacional Autónoma de México

Facultad de Ingeniería

Fundamentos de programación

## **Práctica 12**

Alumna: Rivera González Frida Alison

M.I. Marco Antonio Martínez Quintana

No. lista: 40

No. De equipo empleado: No aplica.

Semestre: 2021-1

Fecha de entrega: viernes 22 de enero de 2021

Grupo: 3

Observaciones:

Calificación: \_\_\_\_\_

## **Práctica 12: Funciones.**

### **Objetivo:**

Elaborar programas en C donde la solución del problema se divida en funciones. Distinguir lo que es el prototipo o firma de una función y la implementación de ella, así como manipular parámetros tanto en la función principal como en otras.

### **Actividades:**

- Implementar en un programa en C la solución de un problema dividido en funciones.
- Elaborar un programa en C que maneje argumentos en la función principal.
- En un programa en C, manejar variables y funciones estáticas.

### **Introducción**

Como ya se mencionó, un programa en lenguaje C consiste en una o más funciones. C permite tener dentro de un archivo fuente varias funciones, esto con el fin de dividir las tareas y que sea más fácil la depuración, la mejora y el entendimiento del código.

En lenguaje C la función principal se llama main. Cuando se ordena la ejecución del programa, se inicia con la ejecución de las instrucciones que se encuentran dentro de la función main, y ésta puede llamar a ejecutar otras funciones, que a su vez éstas pueden llamar a ejecutar a otras funciones, y así sucesivamente.

## Código (Funciones)

```
dena.c amulidi.c amulip.c areglos.c areglos2.c amercado.c funciones.c
#include<stdio.h>
#include<string.h>
/*
Este programa contiene dos funciones: la función main y la función imprimir
. La función main manda llamar a la función imprimir. La función
imprimir recibe como parámetro un arreglo de caracteres y lo recorre de fin a
inicio imprimiendo cada carácter del arreglo.
*/

//Prototipo o firma de las funciones del programa
void imprimir(char[]);

//Definición o implementación de la función main
int main(){
    char nombre[]="Facultad de Ingeniería";
    imprimir(nombre);
}

//Implementación de las funciones del programa
void imprimir(char s[]){
    int tam;
    for( tam=strlen(s)-1 ; tam>=0 ; tam--){
        printf("%c", s[tam]);
        printf("\n");
    }
}
```

```
C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>gcc funciones.c -o funciones.exe

C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>funciones.exe
aireinegni ed datlucaF

C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>_
```

## Código (Ámbito de las variables)

```
ambitofunc.c x amultiap.c x arreglos.c x arreglos2.c x smercado.c x funciones.c x ambitofunc.c x
#include<stdio.h>
/*
Este programa contiene dos funciones: la función main y la función incremento.
La función main manda llamar a la función incremento dentro de un ciclo for.
La función incremento aumenta el valor de la variable enteraGlobal cada
vez que es invocada.
*/

void incremento();
//La variable enteraGlobal es vista por todas
//las funciones (main e incremento)

int enteraGlobal = 0;
int main(){
    //La variable cont es local a la función main
    for (int cont=0 ; cont<5 ; cont++){
        incremento();
    }
    return 999;
}
void incremento () {
    //La variable enterLocal es local a la función incremento
    int enterLocal = 5;
    enteraGlobal += 2;
    printf("global(%i) + local(%i) = %d\n",enteraGlobal,enteraLocal,enteraGlobal+enteraLocal);
}
```

C:\> Símbolo del sistema

```
C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>gcc ambitofunc.c -o ambitofunc.exe

C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>ambitofunc.exe
global(2) + local(5) = 7
global(4) + local(5) = 9
global(6) + local(5) = 11
global(8) + local(5) = 13
global(10) + local(5) = 15

C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>
```

## Código (argumentos función main)

```
#include<stdio.h>
#include<string.h>
/*
Este programa permite manejar los argumentos enviados al ejecutarlo.
*/
int main(int argc,char** argv){
    if (argc==1){
        printf("El programa no contiene argumentos.\n");
        return 88;
    }
    printf("Los elementos del arreglo argv son:\n");
    for(int cont = 0 ; cont<argc ; cont ++){
        printf("argv[%d] = %s\n",cont,argv[cont]);
    }
    return 88;
}
```

```
C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>gcc argumain.c -o argumain.exe
```

```
C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>argumain.exe
```

```
El programa no contiene argumentos.
```

```
C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>_
```

### Código (variable estática)

```
#include<stdio.h>

/*
Este programa contiene dos funciones: la función main y la función
llamarFuncion. La función main manda llamar a la función llamarFuncion
dentro de un ciclo for. La función llamarFuncion crea una variable estática
e imprime su valor.
*/

void llamarFuncion();

int main(){
    for (int j=0; j<5; j++){
        llamarFuncion();
    }
}

void llamarFuncion(){
    static int numVeces = 0;
    printf("Esta funcion se ha llamado %d veces.\n",++numVeces);
}
```

```
C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>gcc varestatica.c -o varestatica.exe

C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>varestatica.exe
Esta funcion se ha llamado 1 veces.
Esta funcion se ha llamado 2 veces.
Esta funcion se ha llamado 3 veces.
Esta funcion se ha llamado 4 veces.
Esta funcion se ha llamado 5 veces.

C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>
```



## Código (función estática)

```
#include<stdio.h>

/*
Este programa contiene las funciones de una calculadora básica:
suma, resta, producto y cociente.
*/

int suma(int,int);
static int resta(int,int);
int producto(int,int);
static int cociente(int,int);

int suma(int a, int b){
    return a+b;
}
static int resta(int a, int b){
    return a-b;
}
int producto(int a, int b){
    return (int) (a*b);
}
static int cociente(int a, int b){
    return(int) (a/b);
}
```

```
1  #include<stdio.h>
2
3  /*
4  Este programa contiene el método principal, el cual
5  invoca a la funciones del archivo funEstatica.c.
6  */
7  int suma(int,int);
8  //static int resta(int,int);
9  int producto(int,int);
10 //static int cociente(int,int);
11
12 int main(){
13     printf("5+7=%i\n",suma(5,7));
14     //printf("9-77=%d\n",resta(9,77));
15     printf("6*8=%i\n",producto(6,8));
16     //printf("7/2=%d\n",cociente(7,2));
17 }
```

```
C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>gcc funcestatica.c calcul.c -o exe
C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>funcestatica.c calcul.c
```

## Conclusión

Los objetivos de la práctica han sido cumplidos en su totalidad, no fue tan difícil de realizar, los ejemplos fueron sencillos permitiendo aún más el entendimiento del tema. Unos ejercicios fueron más claros que otros, en especial el último ejercicio no fue muy claro y no sabía la forma de correrlo, al compilar no me marcaba ningún error, pero no supe cómo hacer que corriera el programa.

## Bibliografía

- Aplicación Notepad
- El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Educación 1991.
- Manual de prácticas de laboratorio.