



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* Marco Antonio Martínez Quintana

*Asignatura:* Fundamentos de programación

*Grupo:* 3

*No de Práctica(s):* 12

*Integrante(s):* Cortés Guadarrama Yair Arturo

*No. de Equipo de  
cómputo empleado:* NO APLICA

*No. de Lista o Brigada:* 11

*Semestre:* 2021-1 (primero)

*Fecha de entrega:* 22/01/2020

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

# 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 (Ámbito de las variables)

```
1  #include <stdio.h>
2  /*
3   Este programa contiene dos funciones: la función main y la función incremento. La
4   función main manda llamar a la función incremento dentro de un ciclo for. La función
5   incremento aumenta el valor de la variable enteraGlobal cada vez que es invocada.
6   */
7  void incremento();
8  // La variable enteraGlobal es vista por todas
9  // las funciones (main e incremento)
10 int enteraGlobal = 0;
11 int main(){
12     // La variable cont es local a la función main
13     for (int cont=0 ; cont<5 ; cont++){
14         incremento();
15     }
16
17     return 999;
18 }
19 void incremento(){
20     // La variable enteraLocal es local a la función incremento
21     int enteraLocal = 5;
22     enteraGlobal += 2;
23     printf("global(%i) + local(%i) = %d\n", enteraGlobal, enteraLocal,
24     enteraGlobal+enteraLocal);
25 }
```

```
C:\Users\yair_\OneDrive\Escritorio\lenguaje c>"practica 12.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\yair_\OneDrive\Escritorio\lenguaje c>■
```

## Código (argumentos función main)

```
1  #include <stdio.h>
2  #include <string.h>
3  /*
4   | Este programa permite manejar los argumentos enviados al ejecutarlo.
5   */
6  int main (int argc, char** argv){
7      if (argc == 1){
8          printf("El programa no contiene argumentos.\n");
9          return 88;
10     }
11
12     printf("Los elementos del arreglo argv son:\n");
13     for (int cont = 0 ; cont < argc ; cont++){
14         printf("argv[%d] = %s\n", cont, argv[cont]);
15     }
16
17     return 88;
18 }
```

```
C:\Users\yair_\OneDrive\Escritorio\lenguaje c>"practica 12.exe"
El programa no contiene argumentos.

C:\Users\yair_\OneDrive\Escritorio\lenguaje c>
```

## 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 función se ha llamado %d veces.\n", ++numVeces);
}
```

```

C:\Users\yair_\OneDrive\Escritorio\lenguaje c>"practica 12.exe"
Esta funci|n se ha llamado 1 veces.
Esta funci|n se ha llamado 2 veces.
Esta funci|n se ha llamado 3 veces.
Esta funci|n se ha llamado 4 veces.
Esta funci|n se ha llamado 5 veces.

C:\Users\yair_\OneDrive\Escritorio\lenguaje c>

```

## Código (función estática)

```

1  //##### funcEstatica.c #####
2  #include <stdio.h>
3  /*
4   Este programa contiene las funciones de una calculadora básica: suma, resta, producto y
5   cociente.
6   */
7  int suma(int,int);
8  static int resta(int,int);
9  int producto(int,int);
10 static int cociente (int,int);
11 int suma (int a, int b){
12     return a + b;
13 }
14 static int resta (int a, int b){
15     return a - b;
16 }
17 int producto (int a, int b){
18     return (int)(a*b);
19 }
20 static int cociente (int a, int b){
21     return (int)(a/b);
22 }
23
1  //##### calculadora.c #####
2  #include <stdio.h>
3  /*
4   Este programa contiene el método principal, el cual invoca a las funciones
5   del archivo funcEstatica.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 int main(){
12     printf("5 + 7 = %i\n",suma(5,7));
13     //printf("9 - 77 = %d\n",resta(9,77));
14     printf("6 * 8 = %i\n",producto(6,8));
15     //printf("7 / 2 = %d\n",cociente(7,2));
16 }

```

```
C:\Users\yair_\OneDrive\Escritorio\lenguaje c>gcc funcEstatica.c calculadora12.c -o calculadora12.exe  
C:\Users\yair_\OneDrive\Escritorio\lenguaje c>calculadora12.exe  
5 + 7 = 12  
6 * 8 = 48  
C:\Users\yair_\OneDrive\Escritorio\lenguaje c>■
```

## **CONCLUSION:**

En esta práctica aprendí a declarar funciones e invocarlas para su uso en un programa y una construcción mas eficiente del mismo para futuros proyectos.

## **REFERENCIAS.**

Guía brindada por la facultad.