



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

Marco Antonio Martínez Quintana

*Profesor:*

Fundamentos de Programación

*Asignatura:*

3

*Grupo:*

Práctica 12

*No de Práctica(s):*

Santiago Díaz Edwin Jaret

*Integrante(s):*

*No. de Equipo de  
cómputo empleado:*

No aplica

50

*No. de Lista o Brigada:*

2021-1

*Semestre:*

Miércoles 20 de Enero del 2021

*Fecha de entrega:*

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

# Estructuras de repetición.

## Objetivos

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.

## Resultados

Código 1:

```
C practico12_1.c > main()
1  #include <stdio.h>
2  #include <string.h>
3
4  /*
5   | Este programa contiene dos funciones: las función main y la función
6   | imprimir. La función main manda llamar a la función imprimir. La función
7   | imprimir recibe como parámetro un arreglo de caracteres y lo recorre de fin a
8   | inicio imprimiendo cada carácter del arreglo.
9   */
10
11 // Prototipo o firma de las funciones del programa
12 void imprimir(char []);
13
14 // Definición o implementación de la función main
15 int main () {
16     char nombre[] = "Facultad de Ingenieria";
17     imprimir(nombre);
18 }
19
20 // Implementación de las funciones del programa
21 void imprimir(char s[]) {
22     int tam;
23     for (tam = strlen(s)-1; tam >= 0 ; tam--)
24         printf("%c", s[tam]);
25     printf("\n");
26 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

1: powershell

```
PS C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamentos de Programación> gcc practico12_1.c -o 12_1
C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamentos de Programación> .\12_1
inegni ed datlucaF
```

## Código 2:

```
C practica12_2.c > incremento()
1  #include <stdio.h>
2
3  /*
4   Este programa contiene dos funciones: la función main y la función incremento. La
5   función main manda llamar a la función incremento dentro de un ciclo for. La función
6   incremento aumenta el valor de la variable enteraGlobal cada vez que es invocada.
7  */
8
9  void incremento();
10 //La variable enteraGlobal es vista por todas
11 // Las funciones (main e incremento)
12
13 int enteraGlobal = 0;
14 int main (){
15     int i;
16     //La variable cont es local a la función main
17     for (i = 0; i < 5; i++)
18     {
19         incremento();
20     }
21     return 999;
22 }
23
24 void incremento() {
25     // La variable enteraLocal es local a la función incremento
26     int enteraLocal = 5;
27     enteraGlobal += 2;
28     printf("global(%i) + local(%i) = %d\n", enteraGlobal, enteraLocal,
29     enteraGlobal+enteraLocal);
30 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

1: powershell



```
PS C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamentos de Programación> gcc practica12_2.c -o 12_2
PS C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamentos de Programación> .\12_2
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
```

### Código 3:

```
C practica12_3.c > main(int, char **)
1  #include <stdio.h>
2  #include <string.h>
3  /*
4   | Este programa permite manejar los argumentos enviados al ejecutarlo
5   */
6
7  int main (int argc, char ** argv) {
8      int i;
9      if (argc == 1)
10     {
11         printf("El programa no conntiene argumentos. ");
12         return 88;
13     }
14
15     printf("Los elementos del arreglo argv son: \n");
16     for (i = 0; i < argc; i++)
17     {
18         printf("argv[%d] = %s\n", i, argv[i]);
19     }
20 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: powershell

```
PS C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamenos de Programación> gcc practica12_3.c -o 12_3
PS C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamenos de Programación> .\12_3
El programa no conntiene argumentos.
```


### Código 4:

```
C practica12_4.c > main()
1  #include <stdio.h>
2
3  /*
4   | Este programa contiene dos funciones: la función main y la función
5   | llamarFunción. La función main manda llamar a la función llamarFunción dentro
6   | de un ciclo for. La función llamarFunción crea una variable estática
7   | e imprime su valor.
8   */
9
10 void llamarFuncion();
11 int i;
12 int main () {
13     for (i = 0; i < 5; i++)
14     {
15         llamarFuncion();
16     }
17 }
18 void llamarFuncion() {
19     static int numVeces = 0;
20     printf("Esta funcion se ha llamado %d veces. \n", ++numVeces);
21 }
```


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: powershell

```
PS C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamenos de Programación> gcc practica12_4.c -o 12_4
PS C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamenos de Programación> .\12_4
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.
```

### Código 5:

```
C practica12_5.c >  cociente(int, int)
1  //#####funcEstatica.c#####
2  #include <stdio.h>
3  /*
4   | Este programa contiene las funciones de una calculadora básica: suma, resta,
5   | producto y cociente
6   */
7
8  int suma(int, int);
9  static int resta(int, int);
10
11 int producto(int, int);
12
13 static int cociente (int, int);
14
15
16 int suma (int a, int b) {
17 |     return a + b;
18 | }
19 static int resta (int a, int b) {
20 |     return a - b;
21 | }
22 int producto (int a, int b) {
23 |     return (int)(a*b);
24 | }
25 static int cociente (int a, int b) {
26 |     return (int)(a/b);
27 | }
```

### Código 6:

```
C practica12_6.c >  main()
1  //##### calculadora.c #####
2  #include <stdio.h>
3
4  /*
5   | Este programa contiene el método principal, el cual invoca a las funciones
6   | del archivo funcEstatica.c En mi caso, practica12_5.c
7   */
8  int suma(int, int);
9  //static int resta(int, int);
10 int producto(int, int);
11 //static int cociente (int, int);
12
13 int main() {
14 |     print("5 + 7 = %i\n", suma(5, 7));
15 |     //print("9 - 77 = %d\n", resta(9,77));
16 |     printf("6 * 8 = %i\n", producto(6,8));
17 |     //print("7 / 2 = %d\n", cociente(7,2));
18 | }
```

## **Conclusiones**

Es mucho más fácil trabajar con funciones, dividir el programa para que así podemos dedicarle a cada función una tarea en específica, así, si hay algún problema en el programa sabremos a donde revisarlo más rápido que antes.