



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:

12

No de Práctica(s):

Zurita León Dana Cecilia

Integrante(s):

*No. de Equipo de
cómputo empleado:*

No aplica

No. de Lista o Brigada:

Primero

Semestre:

18 de Enero de 2021

Fecha de entrega:

Observaciones:

CALIFICACIÓN: _____

Funciones

Objetivo:

Elaborar programas en C donde la solución del problema se divide 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.

Introducción:

Como ya se mencionó, un programa en lenguaje C consiste en una o más funciones.

Dentro de un archivo fuente se pueden tener varias funciones, para poder dividir las tareas y que sea más fácil la depuración y el entendimiento del código. La función principal se llama main. Cuando se ejecuta el 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.

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

```
f Símbolo del sistema
d
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>gcc p12.c -o p12.exe
```

```
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>p12.exe
aireinegnI ed datlucaF
```

```
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>
```

Practica 10.C | Ejemplos.C | Bidimensionales.C | 12.C

```

1 void sumar() {
2     int x;
3     // ámbito de la variable x
4 }
```

Símbolo del sistema

```

C:\Users\dellR\Desktop\LenguajeC\Ejemplos>gcc p121.c -o p121.exe
c:/mingw/bin/.../lib/gcc/mingw32/9.2.0/.../.../mingw32/bin/ld.exe: c:/mingw/bin/.../lib/gcc/mingw32/9.2.0/.../.../lib
mingw32.a(main.o):(.text.startup+0xc0): undefined reference to `WinMain@16'
collect2.exe: error: ld returned 1 exit status
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>
```

```

1 #include <stdio.h>
2 int resultado;
3 void multiplicar() {
4     resultado = 5 * 4;
5 }
```

Símbolo del sistema

```

C:\Users\dellR\Desktop\LenguajeC\Ejemplos>gcc p122.c -o p122.exe
c:/mingw/bin/.../lib/gcc/mingw32/9.2.0/.../.../mingw32/bin/ld.exe: c:/mingw/bin/.../lib/gcc/mingw32/9.2.0/.../.../lib
mingw32.a(main.o):(.text.startup+0xc0): undefined reference to `WinMain@16'
collect2.exe: error: ld returned 1 exit status
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>
```

```

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     return 999;
17 }
18 void incremento(){
19     // La variable enteraLocal es local a la función incremento
20     int enteraLocal = 5;
21     enteraGlobal += 2;
22     printf("global(%i) + local(%i) = %d\n",enteraGlobal, enteraLocal,
23     enteraGlobal+enteraLocal);
24 }
```

ca Símbolo del sistema

```
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>gcc p123.c -o p123.exe  
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>p123.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\dellR\Desktop\LenguajeC\Ejemplos>
```

```
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    return 88;  
17 }
```

ca Símbolo del sistema

```
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>gcc p124.c -o p124.exe  
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>p124.exe  
El programa no contiene argumentos.  
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>
```

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

Símbolo del sistema

```
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>gcc p125.c -o p125.exe
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>p125.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\dellR\Desktop\LenguajeC\Ejemplos>
```

```
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
10 int producto(int,int);
11
12 static int cociente (int,int);
13
14 int suma (int a, int b){
15     return a + b;
16 }
17
18 static int resta (int a, int b){
19     return a - b;
20 }
21 int producto (int a, int b){
22     return (int)(a*b);
23 }
24 static int cociente (int a, int b){
25     return (int)(a/b);
26 }
```

```
C:\ Símbolo del sistema
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>gcc p126.c -o p126.exe
c:/mingw/bin/../lib/gcc/mingw32/9.2.0/../../../../mingw32/bin/ld.exe: c:/mingw/bin/../lib/gcc/mingw32/9.2.0/../../../../lib
mingw32.a(main.o):(.text.startup+0xc0): undefined reference to `WinMain@16'
collect2.exe: error: ld returned 1 exit status
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>
```

```
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
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:\ Símbolo del sistema
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>gcc p127.c -o p127.exe
c:/mingw/bin/../lib/gcc/mingw32/9.2.0/../../../../mingw32/bin/ld.exe: C:\Users\dellR\AppData\Local\Temp\ccmPw2Wp.o:p127.
c:(.text+0x1e): undefined reference to `suma'
c:/mingw/bin/../lib/gcc/mingw32/9.2.0/../../../../mingw32/bin/ld.exe: C:\Users\dellR\AppData\Local\Temp\ccmPw2Wp.o:p127.
c:(.text+0x42): undefined reference to `producto'
collect2.exe: error: ld returned 1 exit status
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>
```

Conclusión:

Después de realizar la práctica pude darme cuenta que se debe determinar el tipo de dato que regresará la función, que las funciones pueden recibir parámetros, que son los que trabaja el programa, estos se definen dentro de paréntesis.

Es fundamental que las funciones se encuentren definidas, también es muy importante el valor de retorno.

Comprendí que las variables locales son las que se declaran dentro de una función y solo existen si la función es llamada.