



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): Ángel Joel Flores Torres

*No. de Equipo de
cómputo empleado:* N/A

No. de Lista o Brigada: 14

Semestre: 1er Semestre

Fecha de entrega: 27 de Enero de 2021

Observaciones:

CALIFICACIÓN: _____

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

Las funciones, también llamados métodos, nos permiten dividir el trabajo que hace un programa, en tareas más pequeñas separadas de la parte principal. Ese es el concepto de función en programación.

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.

RESULTADOS

Código (funciones)

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

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

//Definición o implementación de la función main
int main()
{
    char nombre[]="Facultad de Ingenieria";
    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\angel\Desktop\C>Funciones.exe
aireinegnI ed datlucaF
```

Código (Ámbito de las variables)

```
#include<stdio.h>

void incremento();

//La variables 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 enteraLocal es local a la función incremento
    int enteraLocal=5;
    enteraGlobal+=2;
    printf("global(%i)+local(%i)=%d\n",enteraGlobal,enteraLocal,enteraGlobal+enteraLocal);
}

```

```

C:\Users\angel\Desktop\C>Funciones2.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

```

Código (argumentos función main)

```

#include <stdio.h>
#include <string.h>

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\angel\Desktop\C>Funciones3.exe
El programa no contiene argumentos.

```

Código (variable estática)

```
#include<stdio.h>

void llamarFuncion();

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

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

```
C:\Users\angel\Desktop\C>Funciones4.exe
Esta funcion se ha llamada 1 veces.
Esta funcion se ha llamada 2 veces.
Esta funcion se ha llamada 3 veces.
Esta funcion se ha llamada 4 veces.
Esta funcion se ha llamada 5 veces.
```

Código (función estática)

```
////#####funcEstatica.c#####
#include<stdio.h>

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);
}
```

```

//#####calculadora.c#####
#include<stdio.h>

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

int main()
{
    printf("5+7=%i\n",suma(5,7));
    //printf("9-77=%d\n",resta(9,77));
    printf("6*8=%i\n",producto(6,8));
    //printf("7/2=%d\n",cociente(7,2));
}

```

```

C:\Users\angel\Desktop\C>gcc funcEstatica.c calculadora.c -o exe

C:\Users\angel\Desktop\C>calculadora.exe
"calculadora.exe" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

```

CONCLUSIÓN

Hasta el momento esta práctica y la 10 (que corresponde a depuración) han sido las más complicadas de realizar y entender, pues en este caso son muy pocas las cosas que entendí, los ejemplos que se presentan no son del todo entendibles, pues ya había mencionado con anterioridad que deberían hacer los ejemplos más claros o explicados de una manera más explícitas o incluso más coloquial pues se supone que es fundamentos y hay muchos términos que no entendemos.

REFERENCIAS

¿Qué es una FUNCIÓN (o un método) en programación? (2019). Lenguajes de programación. <https://lenguajesdeprogramacion.net/diccionario/que-es-una-funcion-o-metodo-en-programacion/>