

	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): Rosario Vázquez José André

No. de Equipo de cómputo empleado:

No. de Lista o Brigada: 43

Semestre: Primer semestre

Fecha de entrega: 26 de enero del 2021

Observaciones:

CALIFICACIÓN:

Funciones.

Objetivo:

Elaborar programas en C donde la solución del problema 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.

Objetivo del alumno:

Saber el funcionamiento y la aplicación de las funciones para elaborar un programa en C; además reproducirlo haciendo un programa utilizando funciones.

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.

Actividad.

Comentar los códigos.

```

#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");
}

```

En este código se observa las funciones “ Void Imprimir” y “Void main” donde se ve una estructura de arreglo con palabras y lo recorre desde el principio hasta el fin imprimiendo cada carácter. Como argumento las funciones definidas en el programa no necesariamente deberán ser declaradas; esto dependerá de su ubicación en el código.

Código (Ámbito de variables).

```

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

```

En este código se tienen dos funciones una es “incremento” y la otra “main”, donde su función es incrementar la variable cada vez que es invocada la variable y como dato se declaran fuera de cualquier función se llaman variables globales. Las variables globales existen durante la ejecución de todo el programa y pueden ser utilizadas por cualquier función.

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;
}

```

En este código observo solo una función que la “main”, donde puede recibir como parámetro de entrada un arreglo de cadenas al ejecutar el programa. La longitud del arreglo se guarda en el primer parámetro y el arreglo de cadenas se guarda en el segundo parámetro.

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);
}
```

Se observan dos funciones una “main” y “llamarfuncion”, donde la función llamarfuncion es estática que tiene como característica una variable hace que ésta permanezca en memoria desde su creación y durante toda la ejecución del programa, lo que quiere decir que su valor se mantendrá hasta que el programa llegue a su fin.

Código (función estática)

```
##### funcEstatica.c #####
#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);
}

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

/*
Este programa contiene el método principal, el cual invoca a las funciones
del archivo funcEstatica.c.
*/

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));
}

```

En este caso se están compilando dos códigos al mismo tiempo, donde se observan una función “main” y “estática”, donde las funciones suma y producto son accesibles desde el archivo calculadora y, por tanto, se genera el código ejecutable. Si se quitan los comentarios y se intenta compilar los archivos se enviará un error, debido a que las funciones son estáticas y no pueden ser accedidas fuera del archivo funcEstaticas.c.

Conclusiones.

En conclusión las funciones son un tema complejo; pero estas son sumamente importantes para la realización de programas, donde las funciones permiten automatizar tareas repetitivas, encapsular el código que utilizamos, e incluso mejorar la seguridad, confiabilidad y estabilidad de nuestros programas. Dominar esto es importante ya que permiten modularizar nuestro código, separarlo según las tareas que requerimos.

Referencias.

UNAM. (s. f.). *Laboratorio Salas A y B*. Manual de Prácticas. Recuperado 26 de enero de 2021, de <http://lcp02.fi-b.unam.mx/>