	Carátula para entrega de prácticas
Facultad de Ingeniería	Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor : Alejandro Pimentel

Asignatura: Fundamentos de Programación

Grupo: 3

No de Práctica(s): 9

Integrante(s): Meraz Dionicio Israel

No. de Equipo de cómputo empleado: 9

No. de Lista o Brigada: 8875

Semestre: 1

Fecha de entrega: 14 de Octubre de 2019

Observaciones: Muy bien

CALIFICACIÓN: **10**

OBJETIVO

Elaborar programas en C para la resolución de programas básicos que incluyan las estructuras de repetición y la directiva define.

INTRODUCCION

Cuando decimos que no se puede cambiar hablamos que no se puede cambiar durante la ejecución del programa, es decir, en tiempo de ejecución.

Esa es la principal diferencia entre constante y variable. Una variable puede tener cualquier valor (del mismo tipo de datos que hemos declarado), ya sea en tiempo de diseño, lo cambiamos nosotros en el código fuente, o en tiempo de ejecución, dependiendo de como se está ejecutando el programa.

Pero una constante tendrá su valor inicial, que pondremos en el momento de declararla, siempre. En c se declara así:

```
#define MAX 5
```

EL bucle WHILE se utilizan cuando queremos repetir la ejecución de unas sentencias un número indefinido de veces, siempre que se cumpla una condición.

```
while (expresión_lógica) {
    // Bloque de código a repetir
    // mientras que la expresión
    // lógica sea verdadera.
}
```

El bucle DO-WHILE Se utiliza generalmente cuando no sabemos cuantas veces se habrá de ejecutar el bucle, igual que el bucle WHILE, con la diferencia de que sabemos seguro que el bucle por lo menos se ejecutará una vez.

```
do {
    /*
    Bloque de código que se ejecuta
    por lo menos una vez y se repite
    mientras la expresión lógica sea
    verdadera.
    */
} while (expresión_lógica);
```

EL bucle FOR son estructuras de control cíclicas, que permite ejecutar una o varias lineas de código en forma iterativa. Para que este proceso se dé a cabo, previamente se tiene que asignar un valor de inicio, un valor final y el tamaño de paso.

La principal diferencia entre FOR y WHILE, es que el primero se usa cuando se conoce las veces que va a repetir y en el segundo control cíclico no se conoce el número de repeticiones.

```
for (inicialización ; expresión_lógica ; operaciones por iteración) {
    /*
        Bloque de código
        a ejecutar
    */
}
```

- Hacer un programa que pida un número y muestre su tabla de multiplicar (hasta el 10)

```
#include <stdio.h>

int main() {
    int num;
    int resultado;

    printf("Escibe u numero para la tabla de multiplicacion\n");
    scanf("%i", &num);

    for (int i=1; i<=10; i++) {

        resultado=(int)num*i;
        printf("%i ", num);
        printf("x ");
        printf("%i =", i);
        printf("%i\n", resultado);}


return 0;

}
```

```
Lituania09:Documents fp03alu29$ gcc tabla1.c -o tabla
Lituania09:Documents fp03alu29$ ./tabla
Escibe u numero para la tabla de multiplicacion
5
5 x 1 =5
5 x 2 =10
5 x 3 =15
5 x 4 =20
5 x 5 =25
5 x 6 =30
5 x 7 =35
5 x 8 =40
5 x 9 =45
5 x 10 =50
Lituania09:Documents fp03alu29$
```

- Hacer un programa que pida y lea 10 numeros y muestre su suma y su promedio.

```c
#include <stdio.h>

#define DIV 10

int main (){

    int num;
    float promedio;

    printf("Escribe 10 numeros\n");
    int i=1;
    int z=0;

    while (i<=10) {
        scanf("%i", &num);

        z= num+z;
        i++;

    }
    promedio= (double)z/DIV;
    printf("La suma es de: %i\n", z);
    printf("El promedio es de: %.2lf\n", promedio);

    return 0;

}
```

```
La suma es de: 83
El promedio es de: 8.300000

ACER@EQUIPO /cygdrive/c/Users/ACER/Documents
$ gcc promedio.c -o promedio

ACER@EQUIPO /cygdrive/c/Users/ACER/Documents
$ ./promedio
Escribe 10 numeros
8
4
10
10
9
9
8
8
7
10
La suma es de: 83
El promedio es de: 8.30

ACER@EQUIPO /cygdrive/c/Users/ACER/Documents
$
```

- Hacer un programa que pida un numero e indique si es primo o no.

```c
#include <stdio.h>

int main(){

    int num;

    printf("Ingresa un numero\n");

    int i=1;
    int contador = 0;
    scanf("%i", &num);
    do {

        if (num%i==0){
            contador++;
        }

        i++;

    } while (i<=num);


    if (contador ==2){
        printf("Es primo\n");}
    else {
        printf("No es primo\n");}



    return 0;

}
```

```
ACER@EQUIPO /cygdrive/c/Users/ACER/Documents
$ gcc primo.c -o primo

ACER@EQUIPO /cygdrive/c/Users/ACER/Documents
$ ./primo
Ingresa un numero
3
Es primo

ACER@EQUIPO /cygdrive/c/Users/ACER/Documents
$ ./primo
Ingresa un numero
11
Es primo

ACER@EQUIPO /cygdrive/c/Users/ACER/Documents
$ ./primo
Ingresa un numero
14
No es primo

ACER@EQUIPO /cygdrive/c/Users/ACER/Documents
$
```

CONCLUSION

Se ha observado sobre las diferencias que tiene cada ciclo que son el WHILE, DO-WHILE y el FOR, el while toma primero la condicion y si cumple va a repetir de manera infinita y se va a detener hasta que la condicon sea falsa.
El do-while con diferencia con el while es que evalua la condicion terminando la instrucción, y la instrucción se ejecuta por lo menos una sola vez.
El For a diferencia de los dos anteriores, si conocemos el numero determinado de veces que queremos repeti, y se lee al principio de cada for.
Tambien sobre la utilidad del DEFINE que es la de designar un valor como una constante que no se puede modificar en ningun momento del programa, a diferncia de la variable que si se puede realizar cambios aun cuando le asignes un valor.