	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): Colonia Montero Sonia

*No. de Equipo de
cómputo empleado:* 3

No. de Lista o Brigada: 8

Semestre: 2020-1

Fecha de entrega: 14/10/2019

Observaciones: Bien, pero no cumpliste con el objetivo de utilizar el #define en alguna de las actividades

CALIFICACIÓN: 8

Introducción

Dentro de los códigos de programación, cuando se desea repetir un bloque de código que se ejecute por lo menos una vez mientras cierta expresión lógica sea verdadera, se usan las estructuras de repetición. Éstas pueden ser do-while, while o for.

Por otro lado, existe la directiva define, la cual es una palabra clave para declarar un nombre especial con un significado. Puede ser muy parecido a una variable, con la diferencia de que el valor que se le dé no se podrá cambiar a lo largo del programa.

Ambas de estas estructuras se usaran a lo largo del presente documento con el objetivo de aprender a usarlas correctamente dentro del lenguaje C.

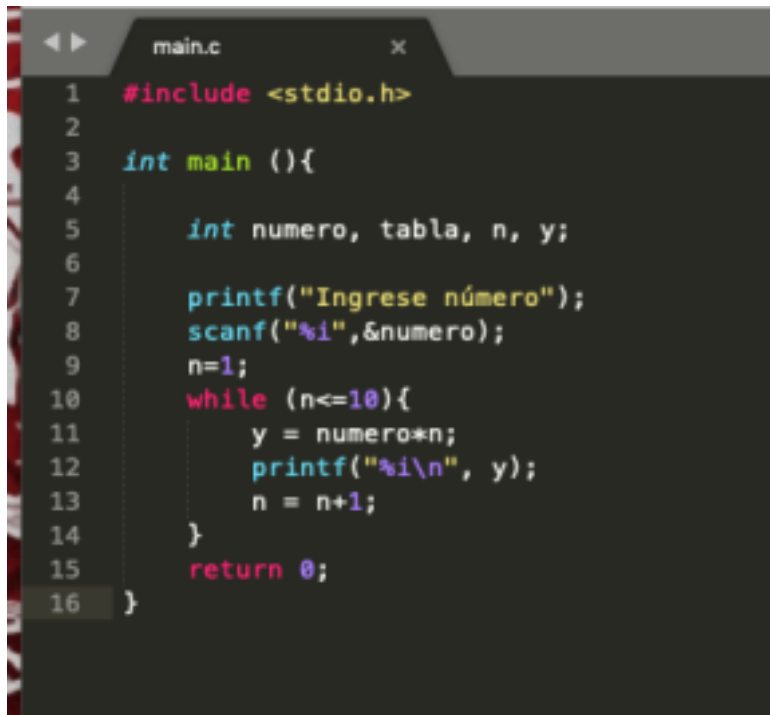
Objetivo

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

Desarrollo

Para cada uno de los siguientes problemas, se eligió un tipo de ciclo y se resolvió. Al final, debieron usar los tres tipos de ciclos y define por lo menos una vez.

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



```
1  #include <stdio.h>
2
3  int main () {
4
5      int numero, tabla, n, y;
6
7      printf("Ingrese número");
8      scanf("%i", &numero);
9      n=1;
10     while (n<=10){
11         y = numero*n;
12         printf("%i\n", y);
13         n = n+1;
14     }
15     return 0;
16 }
```

2. Hacer un programa que pida y lea 10 números y muestre su suma y su promedio.

```

main.c
1  #include <stdio.h>
2  int main () {
3      #define p 10;
4      int n = 0;
5      int m;
6      int y = 0;
7      int promedio;
8      do {
9          printf("Ingrese numero\n");
10         scanf("%i",&m);
11         y = y+m;
12         n = n+1;
13     } while (n<10);
14     promedio = y/p;
15     printf("El promedio es:%i\n", promedio);
16
17     return 0;
18 }

```

3. Hacer un programa que pida un número e indique si es primo o no.

```

main.c  ultimo.c
1  #include <stdio.h>
2  int main(){
3      int numero;
4      int y=0;
5      int n=0;
6      printf("Ingrese número\n");
7      scanf("%i",&numero);
8      for(int d=2; d<numero; d++){
9          n=n+1;
10         if(numero%d!=0){
11             y=y+1;
12         }
13     }
14     if(y==n){
15         printf("Es un número primo\n");
16     }else{
17         printf("No es un número primo\n");
18     }
19     return 0;
20 }
21 }

```

Resultados

Todos los códigos realizados se compilaron y se leyeron para comprobar que se habían realizado correctamente.

Tablas de un número.

```
Laos03:Desktop fp03alu08$ ./main
Ingrese número5
5
10
15
20
25
30
35
40
45
50
Laos03:Desktop fp03alu08$
```

Promedio

```
Lituania09:Desktop fp03alu08$ gcc main.c -o main
Lituania09:Desktop fp03alu08$ ./main
Ingrese numero
1
Ingrese numero
2
Ingrese numero
3
Ingrese numero
4
Ingrese numero
5
Ingrese numero
6
Ingrese numero
7
Ingrese numero
8
Ingrese numero
9
Ingrese numero
10
El promedio es:5
Lituania09:Desktop fp03alu08$
```

Números primos.

```
Lituania09:~ fp03alu08$ cd Desktop
Lituania09:Desktop fp03alu08$ gcc ultimo.c -o ultimo
Lituania09:Desktop fp03alu08$ ./ultimo
4
No es un número primo
Lituania09:Desktop fp03alu08$ ./ultimo
17
Es un número primo
Lituania09:Desktop fp03alu08$ gcc ultimo.c -o ultimo
Lituania09:Desktop fp03alu08$ ./ultimo
Ingrese número
2
Es un número primo
Lituania09:Desktop fp03alu08$ ./ultimo
Ingrese número
3
Es un número primo
Lituania09:Desktop fp03alu08$ ./ultimo
Ingrese número
5
Es un número primo
Lituania09:Desktop fp03alu08$ ./ultimo
Ingrese número
7
Es un número primo
Lituania09:Desktop fp03alu08$ ./ultimo
Ingrese número
11
Es un número primo
Lituania09:Desktop fp03alu08$ ./ultimo
Ingrese número
95
No es un número primo
Lituania09:Desktop fp03alu08$ ./ultimo
Ingrese número
97
Es un número primo
Lituania09:Desktop fp03alu08$
```

Conclusión

Es muy importante aprender a usar las estructuras de repetición en los códigos de programación, ya que permiten hacer acciones que, sin la ayuda de los ciclos que estas estructuras generan, serían muy tediosos, imprecisos o incluso imposibles. Cabe destacar que la directiva define, aunque no es tan indispensable como las estructuras de repetición, si llega a facilitar y hacer más corto el código el uso de ciertos nombres ya estandarizados con cierto valor. Todas estas funciones permiten tener un lenguaje de programación más avanzado y por lo tanto elaborar códigos más complejos y estandarizados.