



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Alejandro Esteban Pimentel Alarcón

Asignatura: Fundamentos de Programación

Grupo: Grupo #3

No de Práctica(s): Práctica No. 9

Integrante(s): Gómez Matias Paola Donaji / Laureano González David / Ortiz Luciano Gerson Gael

*No. de Equipo de
cómputo empleado:*

No. de Lista o Brigada: Número de lista (Por orden de nombre): 17 / 25 / 38
No. de Cuenta (Por orden de nombre): #9472 / #9519 / #1949

Semestre: 2020-1

Fecha de entrega: 14 de Octubre de 2019

Observaciones: No cumplieron con el objetivo de utilizar los tres tipos de ciclos. Esto es lo que más baja su calificación. Además, en la segunda actividad hicieron lectura hacia muchas variables en lugar de poner un ciclo que leyera muchas veces, ese no era el objetivo de la actividad.

CALIFICACIÓN: 7

Introducción

Estructuras de repetición

Las estructuras de repetición hacen posible la ejecución repetida de una o más instrucciones. Estas estructuras describen procesos que se repiten varias veces para la solución de problemas.

- **While**

Esta estructura se encarga de repetir de forma cíclica un conjunto de instrucciones que se encuentren dentro del bucle, esta repetición se repetirá hasta que se cumpla la condición que definamos

- En C

```
while (condicion de parada)
{
    // Instrucciones del ciclo
}
```

- En diagrama de flujo



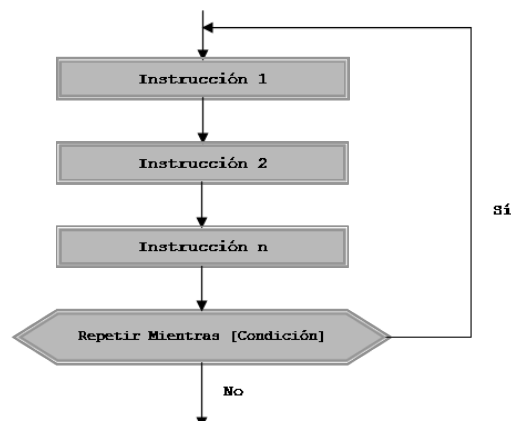
- **Do – While**

Son una estructura de control cíclica, los cuales nos permiten ejecutar una o varias líneas de código de forma repetitiva sin necesidad de tener un valor inicial e incluso a veces sin siquiera conocer cuando se va a dar el valor final. El ciclo do-while nos da la posibilidad de ejecutar primero el bloque de instrucciones antes de evaluar la condición necesaria, de este modo los ciclos do-while, son más efectivos para algunas situaciones específicas.

- En C

```
do
{
    ....
    ....
    Bloque de Instrucciones....
    ....
    ....
}
while(condición de finalización);
```

- En diagrama de flujo



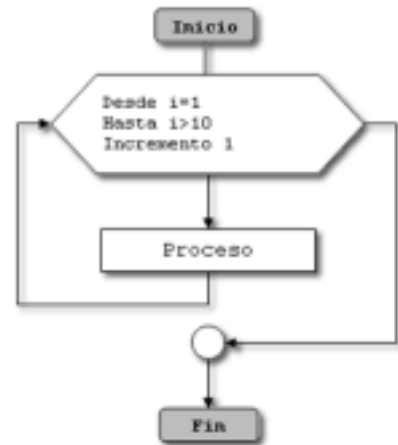
- **For**

Es un ciclo que repite un conjunto de instrucciones mientras una condición es verdadera y a diferencia de los demás ciclos, en este sabemos el número de veces que se repetirá el ciclo

➤ En C

```
for (Expresión-de-inicio; Condición; incremento)
{
    Instrucciones del ciclo.
}
```

➤ En diagrama de flujo



- **#define**

El define se utiliza para declarar un nombre especial con un significado. Es muy parecido a una variable pero con la diferencia que esta no se puede cambiar a lo largo del programa.

Objetivo

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

Actividad No. 1


Hicimos un programa que pide un número y muestra sus tablas de multiplicar hasta el número 10

```
main.c
1  #define DIEZ 10
2  #include <stdio.h>
3  int main()
4  {
5      char z;
6      int i, x;
7      do
8      {
9          printf("\n Introduzca un numero entero \n");
10         scanf("%i", &x);
11         printf("\n La tabla de multiplicar de %i es: \n", x);
12         for(i=1;i<=DIEZ;i++)
13         {
14             printf("\n %i * %i = %i", i, x, i*x);
15         }
16     }while(z!=0);
17
18     return 0;
19 }
```

Lo compilamos y lo corremos y nos pide que introduzcamos un número



Introducimos el número y nos muestra su table de multiplicar hasta el 10



```
Introduzca un numero entero
8
La tabla de multiplicar de 8 es:
1 * 8 = 8
2 * 8 = 16
3 * 8 = 24
4 * 8 = 32
5 * 8 = 40
6 * 8 = 48
7 * 8 = 56
8 * 8 = 64
9 * 8 = 72
10 * 8 = 80
```

Actividad No. 2

Hicimos un programa que pida 10 números y muestre su suma y su promedio

```

main.c  (*) main2.c
2      #define Y 10
3      int main()
4      {
5          char resp,opt;
6          float v,.,z,z,t,t,c,c,c,c,d,g,p,i,s,j;
7          do
8          {
9              printf("Promedios");
10             printf("\n\n Inserte los valores(10) para sacar el promedio. '\n\n'");
11             printf("Inserte el primer valor:\n\n");
12             scanf("%f", &i);
13             printf("Inserte el segundo valor:\n\n");
14             scanf("%f", &t);
15             printf("Inserte el tercer valor:\n\n");
16             scanf("%f", &c);
17             printf("Inserte el cuarto valor:\n\n");
18             scanf("%f", &c);
19             printf("Inserte el quinto valor:\n\n");
20             scanf("%f", &c);
21             printf("Inserte el sexto valor:\n\n");
22             scanf("%f", &t);
23             printf("Inserte el septimo valor:\n\n");
24             scanf("%f", &g);
25             printf("Inserte el octavo valor:\n\n");
26             scanf("%f", &p);
27             printf("Inserte el noveno valor:\n\n");
28             scanf("%f", &i);
29             printf("Inserte el decimo valor:\n\n");
30             scanf("%f", &j);
31             v+=c+c+c+c+c+c+c+c+c+c;
32             z+=v;
33             printf("La suma de los valores antes ingresado en %f y el promedio en %f.",v,z);
34             printf("\n\n ¿Desea sacar otro promedio ? '\n\n'");
35             printf("\n\n      S: Si                      N: No '\n\n'");
36             scanf("%c", &opt);
37         }
38         while (opt != 'N');
39         printf("\n\n ¡Masta luego! '\n\n'");
40         return 0;
41     }
42
43
44

```

El objetivo del ciclo no era que el proceso completo se repitiera, sino que el proceso de leer se hiciera varias veces en una misma línea de código. No hacía falta mas que una variable

Lo compilamos y corremos y nos pide insertar los 10 valores de los cuales va a sacar su promedio y a hacer su suma

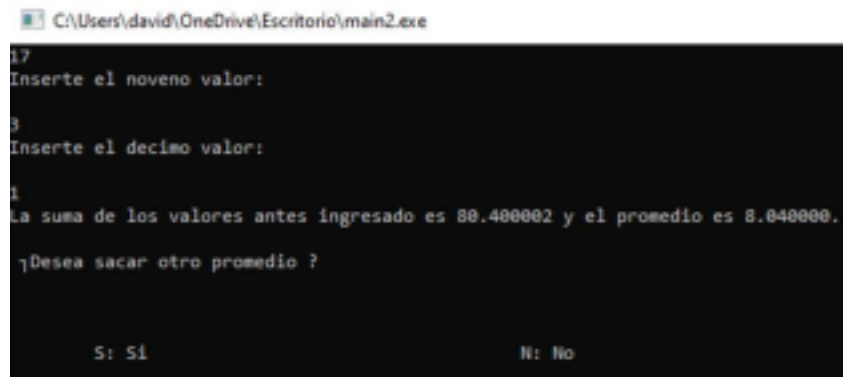
```
C:\Users\David\OneDrive\Escritorio\main2.exe
Promedios

Inserte los valores(10) para sacar el promedio.

Inserte el primer valor:
6
Inserte el segundo valor:
7
Inserte el tercer valor:
9
Inserte el cuarto valor:
15
Inserte el quinto valor:
9.23
Inserte el sexto valor:
5.17
Inserte el septimo valor:
8
Inserte el octavo valor:
17
Inserte el noveno valor:
```

```
C:\Users\david\OneDrive\Escritorio\main2.exe
17
Inserte el noveno valor:
3
Inserte el decimo valor:
1_
```

Una vez que tenemos los diez valores nos muestra su promedio y su suma

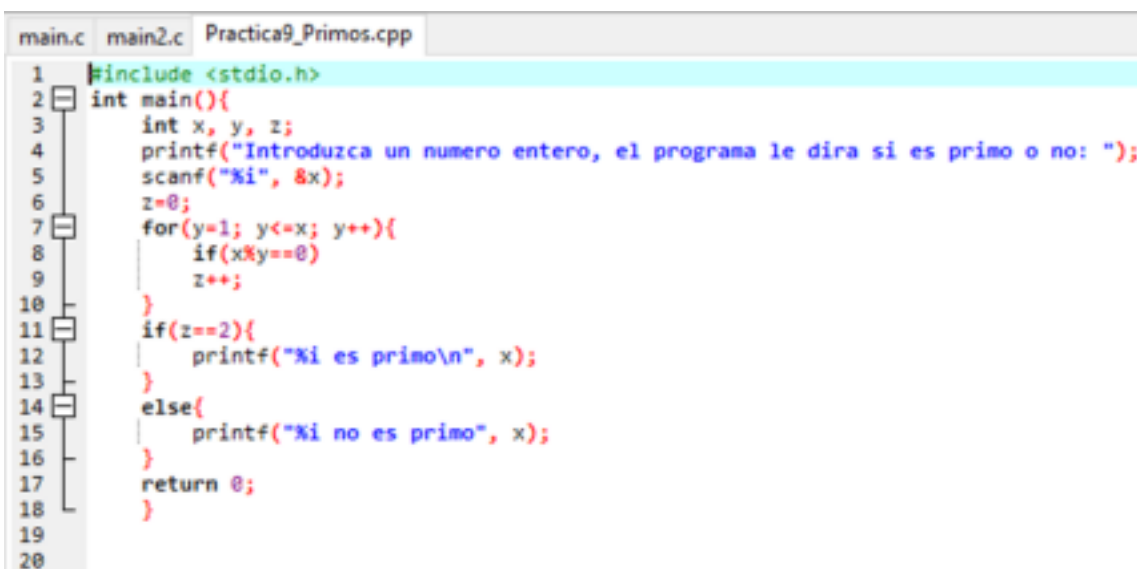


```
C:\Users\david\OneDrive\Escritorio\main2.exe
17
Inserte el noveno valor:
3
Inserte el decimo valor:
1
La suma de los valores antes ingresado es 80.400002 y el promedio es 8.040000.
¿Desea sacar otro promedio ?

S: Si      N: No
```

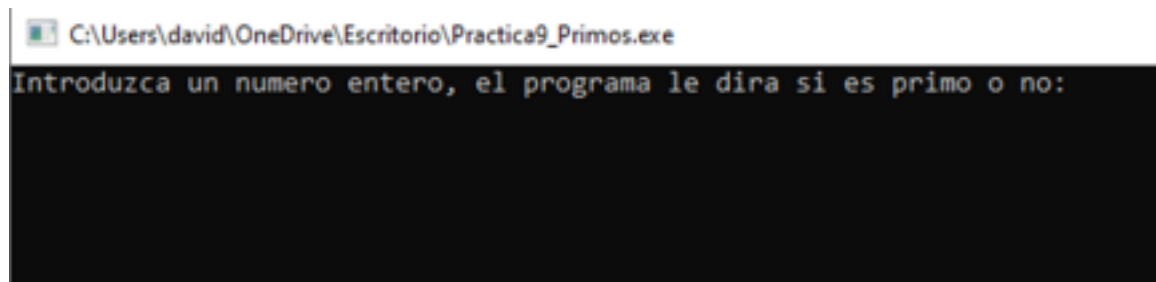
Actividad No. 3

Hicimos un programa que pide un número e indica si este es primo o no



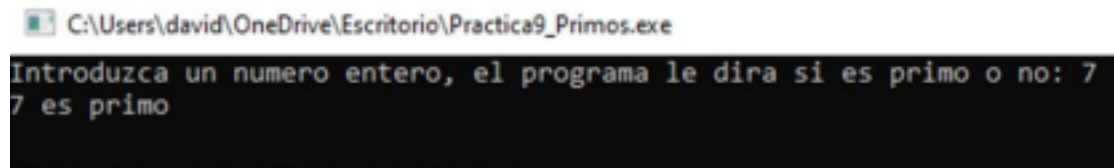
```
main.c  main2.c  Practica9_Primos.cpp
1  #include <stdio.h>
2  int main(){
3      int x, y, z;
4      printf("Introduzca un numero entero, el programa le dira si es primo o no: ");
5      scanf("%i", &x);
6      z=0;
7      for(y=1; y<=x; y++){
8          if(x%y==0)
9              z++;
10     }
11     if(z==2){
12         printf("%i es primo\n", x);
13     }
14     else{
15         printf("%i no es primo", x);
16     }
17     return 0;
18 }
19
20
```

Lo compilamos y lo corremos y nos pedirá un número



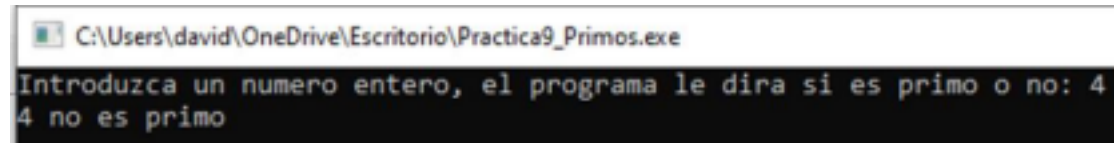
```
C:\Users\david\OneDrive\Escritorio\Practica9_Primos.exe
Introduzca un numero entero, el programa le dira si es primo o no:
```

Introducimos un número y si es primo



```
C:\Users\david\OneDrive\Escritorio\Practica9_Primos.exe
Introduzca un numero entero, el programa le dira si es primo o no: 7
7 es primo
```

Y si no es primo



```
C:\Users\david\OneDrive\Escritorio\Practica9_Primos.exe
Introduzca un numero entero, el programa le dira si es primo o no: 4
4 no es primo
```

Conclusiones

Gracias a estas estructuras de repetición, aprendimos nuevas herramientas para hacer programas las cuales podemos aplicar en los desarrollos de futuros proyectos, gracias a los ciclos for, do y do-while. También con el #define, aprendimos que podemos tener una variable con un valor fijo para usarla durante todo el programa y que gracias a que no puede cambiar de valor, se puede utilizar de distintas formas con su valor fijo.