	Carátula para entrega de prácticas	
Facultad de Ingeniería	Laboratorio de docencia	

Laboratorios de computación salas A y B

Profesor: ALEJANDRO ESTEBAN PIMENTEL ALARCON

Asignatura: FUNDAMENTOS DE PROGRAMACION

Grupo: 03

No de Práctica(s): PRACTICA No. 9

Integrante(s): MORONES FLORES INGRID YOHUALLI

*No. de Equipo de
cómputo empleado:*

No. de Lista o Brigada:

Semestre: 2020-1

Fecha de entrega: 14/OCTUBRE/2019

Observaciones: Bien, pero mejora esta carátula, no debe de pasar una hoja.

CALIFICACIÓN: 10

ESTRUCTURAS DE REPETICION.

OBJETIVO.

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

INTRODUCCION.

Recuerda que lo que estamos usando es C. Una Estructura de Repetición en C++ le permite al programador especificar que se repita una acción, en tanto cierta condición se mantenga verdadera.

while

La estructura de repetición While ejecuta un ciclo que se repetirá mientras que la condición sea verdadera. Sintaxis while (<condición>) Ejemplo: while (a <= 10)

Do/while

La estructura do/while es similar a la estructura while. En la while, la condición de continuación de ciclo se prueba al principio del ciclo, antes de ejecutarse el cuerpo del mismo. La estructura do/while prueba la condición de continuación del ciclo repetitivo, después de ejecutar el cuerpo del ciclo, por lo tanto, el cuerpo del ciclo repetitivo se ejecutará por lo menos una vez.

Cuando termina do/while, la ejecución continuará con el enunciado que aparezca después de la cláusula while. No es necesario utilizar llaves

Sintaxis do Sentencias while (<condicion>); Ejemplo do { cout <<"entre la nota"; cin>>nota; i++; } while(i <=10);

For

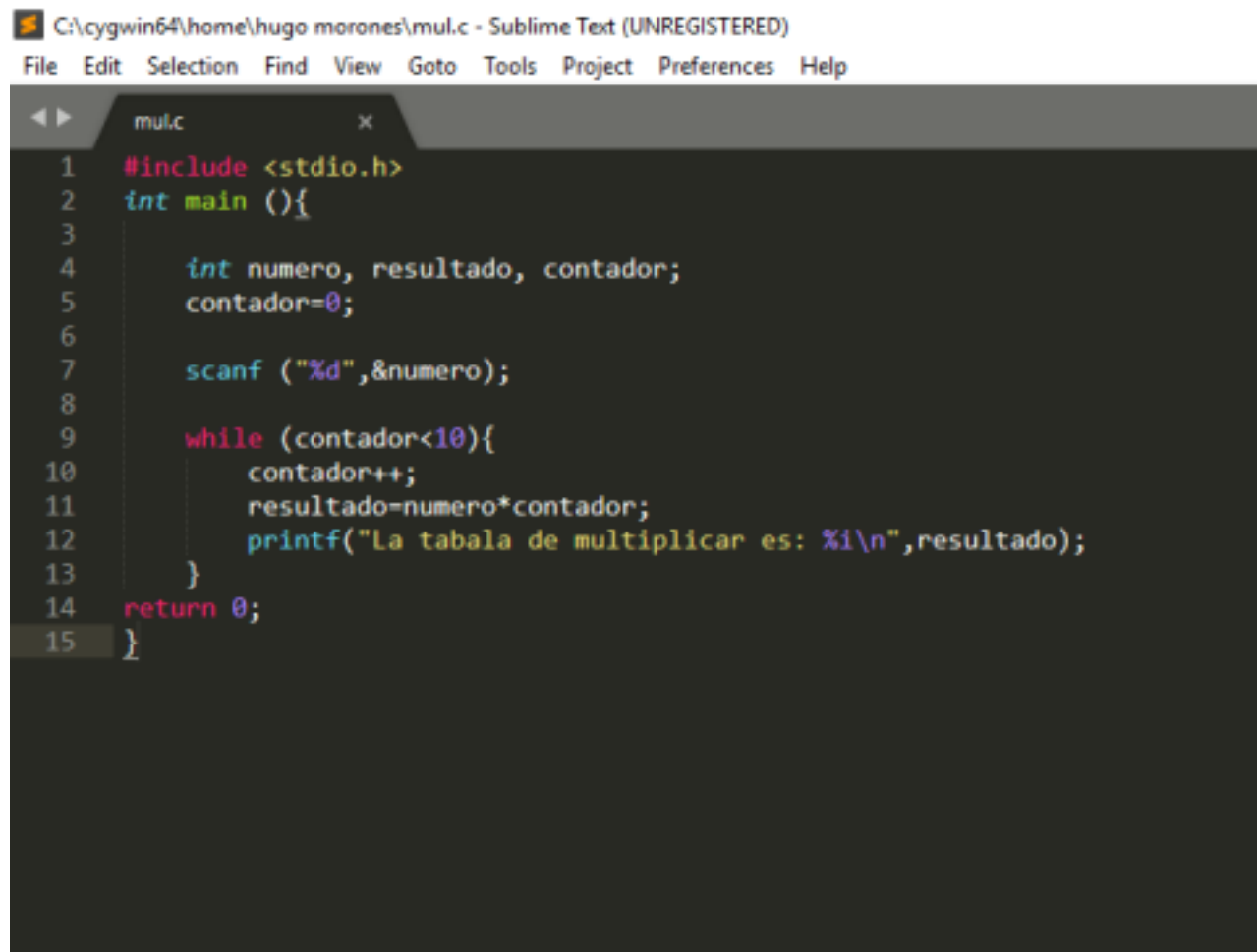
La estructura de repetición for manera de manera automática todos los detalles de la repetición controlada por contador. Sintaxis for (<inicio;final;contador>) Ejemplo: for(int i = 0;i <= 10;i++) cout <<"hola";

Define.

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

ACTIVIDAD 1.

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



The image shows a screenshot of a C program named 'mul.c' in the Sublime Text editor. The program is designed to take an integer input from the user and print its multiplication table up to 10. The code is as follows:

```
1  #include <stdio.h>
2  int main () {
3
4      int numero, resultado, contador;
5      contador=0;
6
7      scanf ("%d",&numero);
8
9      while (contador<10){
10         contador++;
11         resultado=numero*contador;
12         printf("La tabala de multiplicar es: %i\n",resultado);
13     }
14     return 0;
15 }
```



```
La tabala de multiplicar es: 81  
La tabala de multiplicar es: 90
```

```
hugo morones@LAPTOP-FGPEUFLH ~
```

```
$ ./multi
```

```
10
```

```
La tabala de multiplicar es: 10  
La tabala de multiplicar es: 20  
La tabala de multiplicar es: 30  
La tabala de multiplicar es: 40  
La tabala de multiplicar es: 50  
La tabala de multiplicar es: 60  
La tabala de multiplicar es: 70  
La tabala de multiplicar es: 80  
La tabala de multiplicar es: 90  
La tabala de multiplicar es: 100
```

```
hugo morones@LAPTOP-FGPEUFLH ~
```

```
$ gcc mul.c -o multi
```

```
hugo morones@LAPTOP-FGPEUFLH ~
```

```
$ ./multi
```

```
6
```

```
La tabala de multiplicar es: 6  
La tabala de multiplicar es: 12  
La tabala de multiplicar es: 18  
La tabala de multiplicar es: 24  
La tabala de multiplicar es: 30  
La tabala de multiplicar es: 36  
La tabala de multiplicar es: 42  
La tabala de multiplicar es: 48  
La tabala de multiplicar es: 54  
La tabala de multiplicar es: 60
```

```
hugo morones@LAPTOP-FGPEUFLH ~
```

```
$ ./multi
```

```
5
```

```
La tabala de multiplicar es: 5  
La tabala de multiplicar es: 10  
La tabala de multiplicar es: 15  
La tabala de multiplicar es: 20  
La tabala de multiplicar es: 25  
La tabala de multiplicar es: 30  
La tabala de multiplicar es: 35  
La tabala de multiplicar es: 40  
La tabala de multiplicar es: 45  
La tabala de multiplicar es: 50
```

```
hugo morones@LAPTOP-FGPEUFLH ~
```

```
$
```



Escribe aquí para buscar





```
$ gcc mul.c -o multi
```

```
hugo morones@LAPTOP-FGPEUFLH ~
```

```
$ ./multi
```

```
2
```

```
La tabala de multiplicar es: 2  
La tabala de multiplicar es: 4  
La tabala de multiplicar es: 6  
La tabala de multiplicar es: 8  
La tabala de multiplicar es: 10  
La tabala de multiplicar es: 12  
La tabala de multiplicar es: 14  
La tabala de multiplicar es: 16  
La tabala de multiplicar es: 18  
La tabala de multiplicar es: 20
```

```
hugo morones@LAPTOP-FGPEUFLH ~
```

```
$ ./multi
```

```
9
```

```
La tabala de multiplicar es: 9  
La tabala de multiplicar es: 18  
La tabala de multiplicar es: 27  
La tabala de multiplicar es: 36  
La tabala de multiplicar es: 45  
La tabala de multiplicar es: 54  
La tabala de multiplicar es: 63  
La tabala de multiplicar es: 72  
La tabala de multiplicar es: 81  
La tabala de multiplicar es: 90
```

```
hugo morones@LAPTOP-FGPEUFLH ~
```

```
$ ./multi
```

```
10
```

```
La tabala de multiplicar es: 10  
La tabala de multiplicar es: 20  
La tabala de multiplicar es: 30  
La tabala de multiplicar es: 40  
La tabala de multiplicar es: 50  
La tabala de multiplicar es: 60  
La tabala de multiplicar es: 70  
La tabala de multiplicar es: 80  
La tabala de multiplicar es: 90  
La tabala de multiplicar es: 100
```

```
hugo morones@LAPTOP-FGPEUFLH ~
```

```
$ gcc mul.c -o multi
```

```
hugo morones@LAPTOP-FGPEUFLH ~
```

```
$ ./multi
```

```
6
```

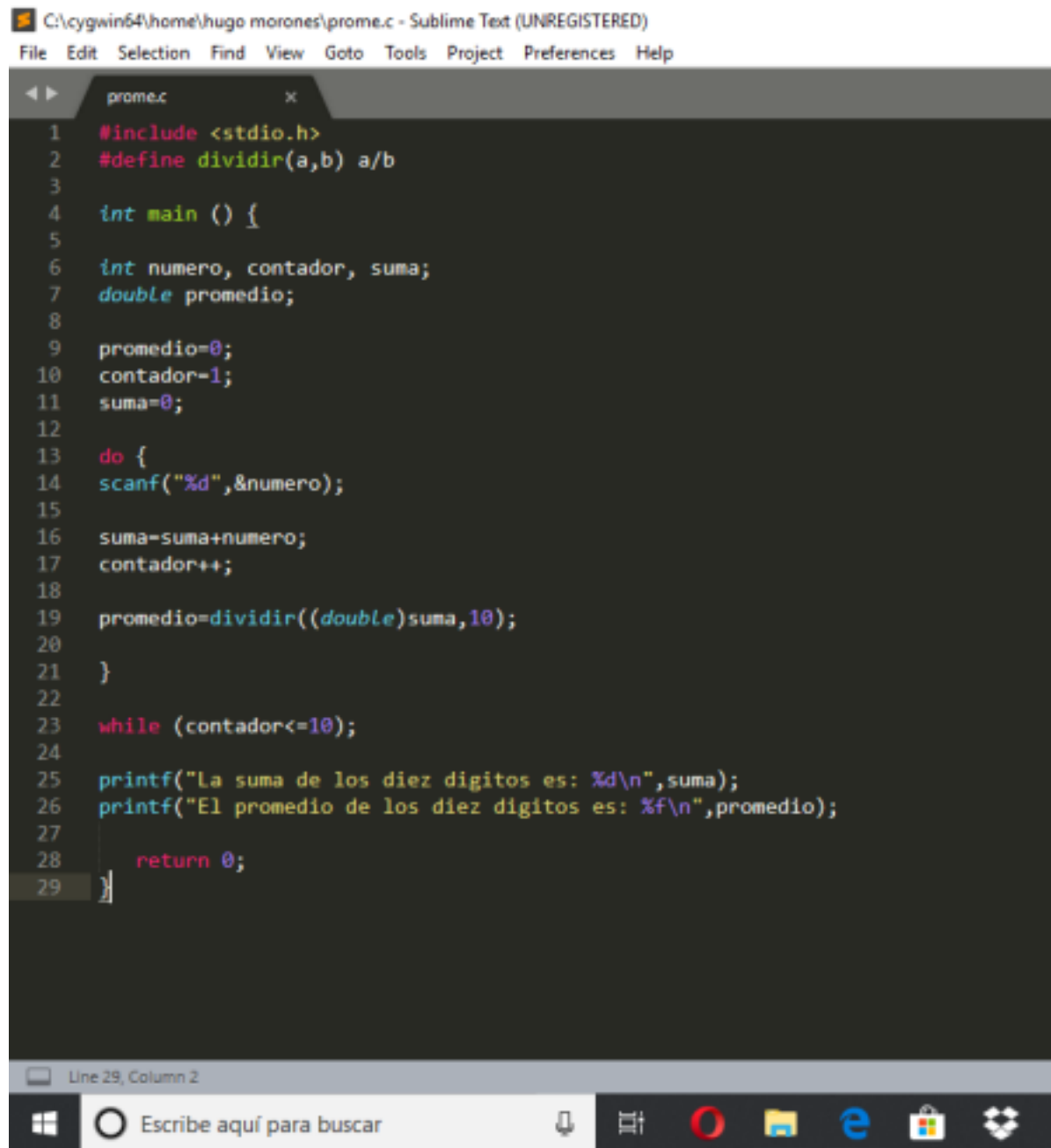


Escribe aquí para buscar



ACTIVIDAD 2.

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



The screenshot shows a Sublime Text editor window titled "C:\cygwin64\home\hugo morones\promed.c - Sublime Text (UNREGISTERED)". The menu bar includes File, Edit, Selection, Find, View, Goto, Tools, Project, Preferences, and Help. The editor displays a C program named "promed.c" with the following code:

```
1  #include <stdio.h>
2  #define dividir(a,b) a/b
3
4  int main () {
5
6      int numero, contador, suma;
7      double promedio;
8
9      promedio=0;
10     contador=1;
11     suma=0;
12
13     do {
14         scanf("%d",&numero);
15
16         suma=suma+numero;
17         contador++;
18
19         promedio=dividir((double)suma,10);
20
21     }
22
23     while (contador<=10);
24
25     printf("La suma de los diez digitos es: %d\n",suma);
26     printf("El promedio de los diez digitos es: %f\n",promedio);
27
28     return 0;
29 }
```

The status bar at the bottom indicates "Line 29, Column 2". The Windows taskbar is visible at the bottom with the search bar containing "Escribe aquí para buscar" and several application icons.



```
hugo morones@LAPTOP-FGPEUFLH ~  
$ gcc prome.c -o promedio  
  
hugo morones@LAPTOP-FGPEUFLH ~  
$ ./promedio  
3  
5  
6  
9  
3  
6  
2  
9  
6  
4  
La suma de los diez digitos es: 53  
El promedio de los diez digitos es: 5.300000
```

```
hugo morones@LAPTOP-FGPEUFLH ~  
$ ./promedio  
4  
4  
5  
5  
6  
6  
8  
8  
2  
2  
La suma de los diez digitos es: 50  
El promedio de los diez digitos es: 5.000000
```

```
hugo morones@LAPTOP-FGPEUFLH ~  
$ ./promedio  
5  
7  
5  
7  
5  
7  
5  
7  
La suma de los diez digitos es: 60  
El promedio de los diez digitos es: 6.000000
```

```
hugo morones@LAPTOP-FGPEUFLH ~
```



Escribe aquí para buscar



ACTIVIDAD 3.

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



The screenshot shows a Sublime Text editor window titled "C:\cygwin64\home\hugo morones\prim.c - Sublime Text (UNREGISTERED)". The menu bar includes File, Edit, Selection, Find, View, Goto, Tools, Project, Preferences, and Help. The editor displays a C program named "prim.c" with the following code:

```
1  #include <stdio.h>
2  int main () {
3
4  int i, numero, contador;
5  contador=0;
6
7  scanf ("%i",&numero);
8  for (i=1; i<=numero; i++) {
9      if (numero%i==0){
10         contador++;
11     }
12 }
13 if (contador>2){
14     printf("Compuesto\n");
15 }
16 else {
17     printf("Primo\n");
18 }
19 return 0;
20 }
```



```
hugo morones@LAPTOP-FGPEUFLH ~  
$ pwd  
/home/hugo morones  
  
hugo morones@LAPTOP-FGPEUFLH ~  
$ ls  
prim.c vim  
  
hugo morones@LAPTOP-FGPEUFLH ~  
$ gcc prim.c -o primo  
  
hugo morones@LAPTOP-FGPEUFLH ~  
$ ./primo  
5  
Primo  
  
hugo morones@LAPTOP-FGPEUFLH ~  
$ ./primo  
2  
Primo  
  
hugo morones@LAPTOP-FGPEUFLH ~  
$ ./primo  
6  
Compuesto  
  
hugo morones@LAPTOP-FGPEUFLH ~  
$ ./primo  
90  
Compuesto  
  
hugo morones@LAPTOP-FGPEUFLH ~  
$ ./primo  
83  
Primo  
  
hugo morones@LAPTOP-FGPEUFLH ~  
$ |
```

CONCLUSIONES.

Pudimos llegar con éxito al objetivo realizando programas que contuvieran las estructuras de repetición, siempre tomando en cuenta lo que se nos pedía.