



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): CARLOS SOTELO LEYVA

*No. de Equipo de
cómputo empleado:* 47

No. de Lista o Brigada: 47

Semestre: 2020-1

Fecha de entrega: 14/10/2019

Observaciones: Bien, tienes un error en tu último código, pero más importante que eso, no cumpliste con usar

#define en alguna de tus actividades

CALIFICACIÓN: 8

C es un lenguaje de programación de propósito general que ofrece economía sintáctica, control de flujo y estructuras sencillas y un buen conjunto de operadores. No es un lenguaje de muy alto nivel y más bien un lenguaje pequeño, sencillo y no está especializado en ningún tipo de aplicación. Esto lo hace un lenguaje potente, con un campo de aplicación ilimitado y sobre todo, se aprende rápidamente. En poco tiempo, un programador puede utilizar la totalidad del lenguaje.

Uno de los objetivos de diseño del lenguaje C es que solo sean necesarias unas pocas instrucciones en lenguaje máquina para traducir cada elemento del lenguaje, sin que haga falta un soporte intenso en tiempo de ejecución. Es muy posible escribir C a bajo nivel de abstracción; de hecho, C se usó como intermediario entre diferentes lenguajes. Uno de las principales funciones de el son hacer programas complejos como los que estaremos viendo a continuación.

OBJETIVO DE LA PRÁCTICA:

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

WHILE:

```
While (expression_lógica) {  
    // bloque de código a repetir  
    // mientras que la expression  
    // lógica sea verdadera.  
}
```

Como lo dice esta expression nos sirve para que cuando estemos hacienda un programa se repita lo que está dentro de este.

DO WHILE:

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

Esta expresión es parecida a la de “while” ya que también este se repite por lo menos una vez dentro del programa mientras que lo que se de es verdadero.

FOR:

```
For (inicialización ; expresión lógica ; operaciones por interacción) {  
    /*  
        Bloque de código  
        A ejecutar  
    */  
}
```

Esta expresión nos sirve para hacer operaciones.

DEFINE:

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

Se escribe como:

#define MAX 5

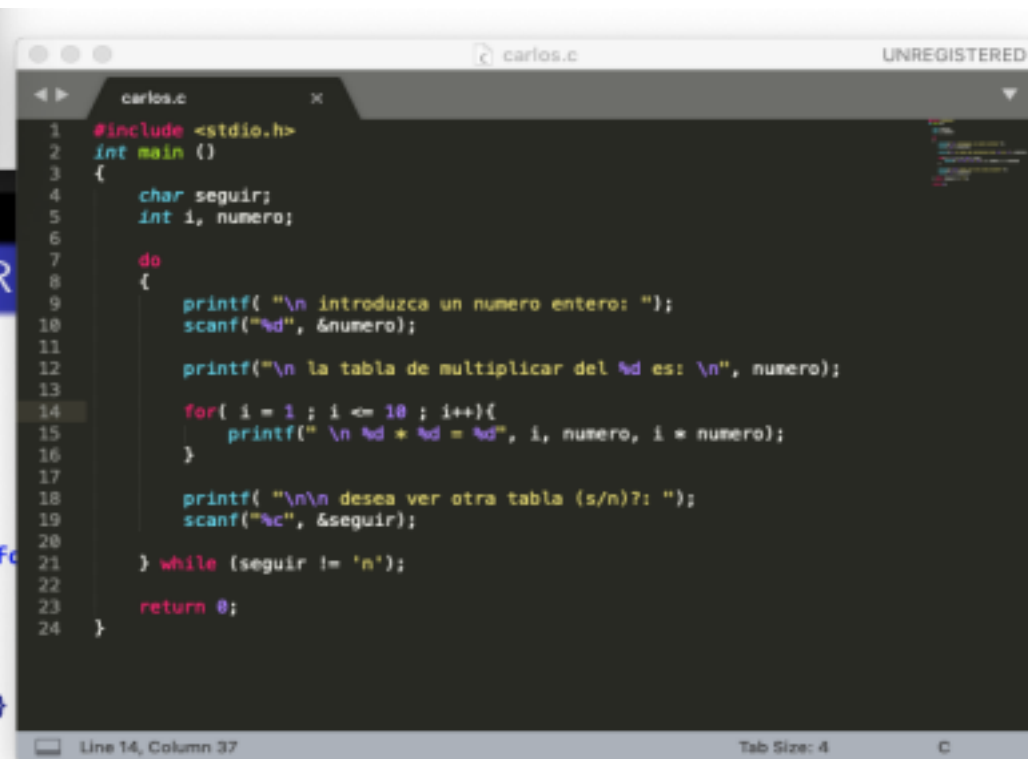
ACTIVIDADES:

Para cada uno de los siguientes problemas, elegir un tipo de ciclo y resolverlo. Al final, deben usar los tres tipos de ciclos y usar define por lo menos una vez.

- Hacer un programa que pida un número y muestre su tabla de multiplicar (hasta el 10).
- Hacer un programa que pida y lea 10 números y muestre su suma y su promedio.
- Hacer un programa que pida un número e indique si es primo o no.

Actividad 1:

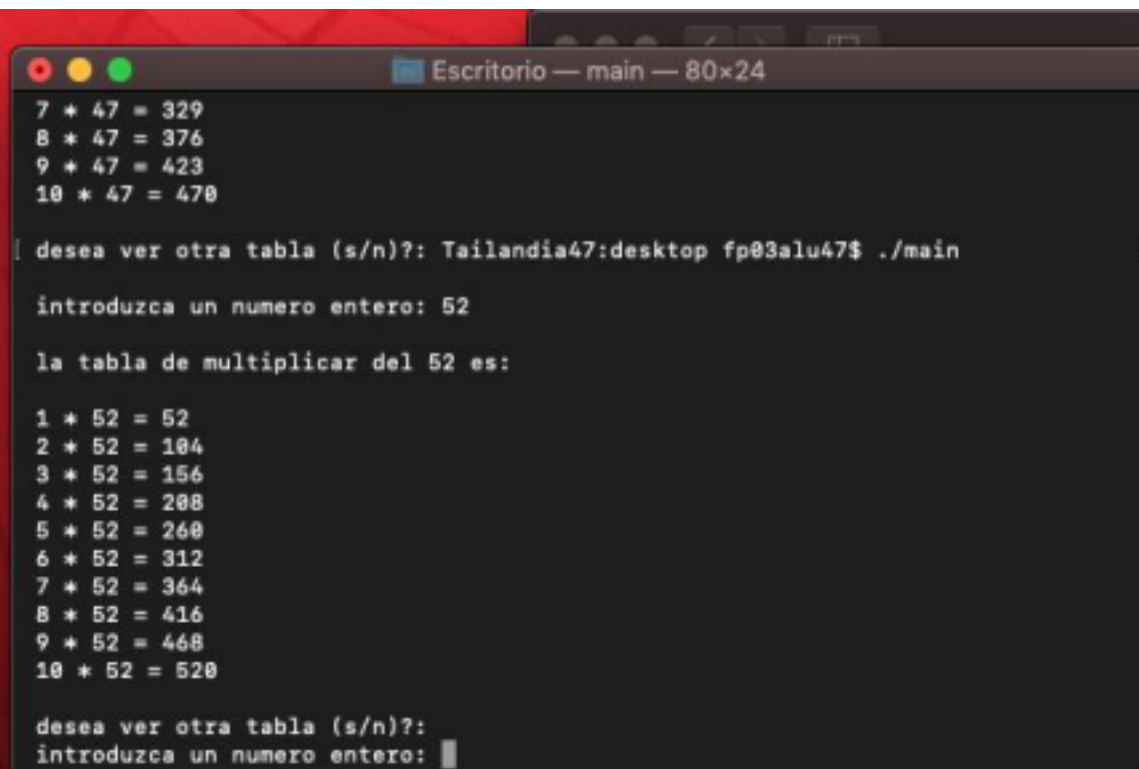
Hacer un programa que pida un número y muestre su tabla de multiplicar (hasta el 10).



The screenshot shows a code editor window titled 'carlos.c' with a 'UNREGISTERED' label in the top right corner. The code is written in C and implements a program that asks for a number and prints its multiplication table. The code is as follows:

```
1  #include <stdio.h>
2  int main ()
3  {
4      char seguir;
5      int i, numero;
6
7      do
8      {
9          printf( "\n introduzca un numero entero: ");
10         scanf("%d", &numero);
11
12         printf("\n la tabla de multiplicar del %d es: \n", numero);
13
14         for( i = 1 ; i <= 10 ; i++){
15             printf(" \n %d * %d = %d", i, numero, i * numero);
16         }
17
18         printf( "\n\n desea ver otra tabla (s/n)?: ");
19         scanf("%c", &seguir);
20
21     } while (seguir != 'n');
22
23     return 0;
24 }
```

The status bar at the bottom indicates 'Line 14, Column 37', 'Tab Size: 4', and 'C'.



The screenshot shows a terminal window titled 'Escritorio — main — 80x24'. The output of the program is as follows:

```
7 * 47 = 329
8 * 47 = 376
9 * 47 = 423
10 * 47 = 470

[ desea ver otra tabla (s/n)?: Tailandia47:desktop fp03alu47$ ./main

introduzca un numero entero: 52

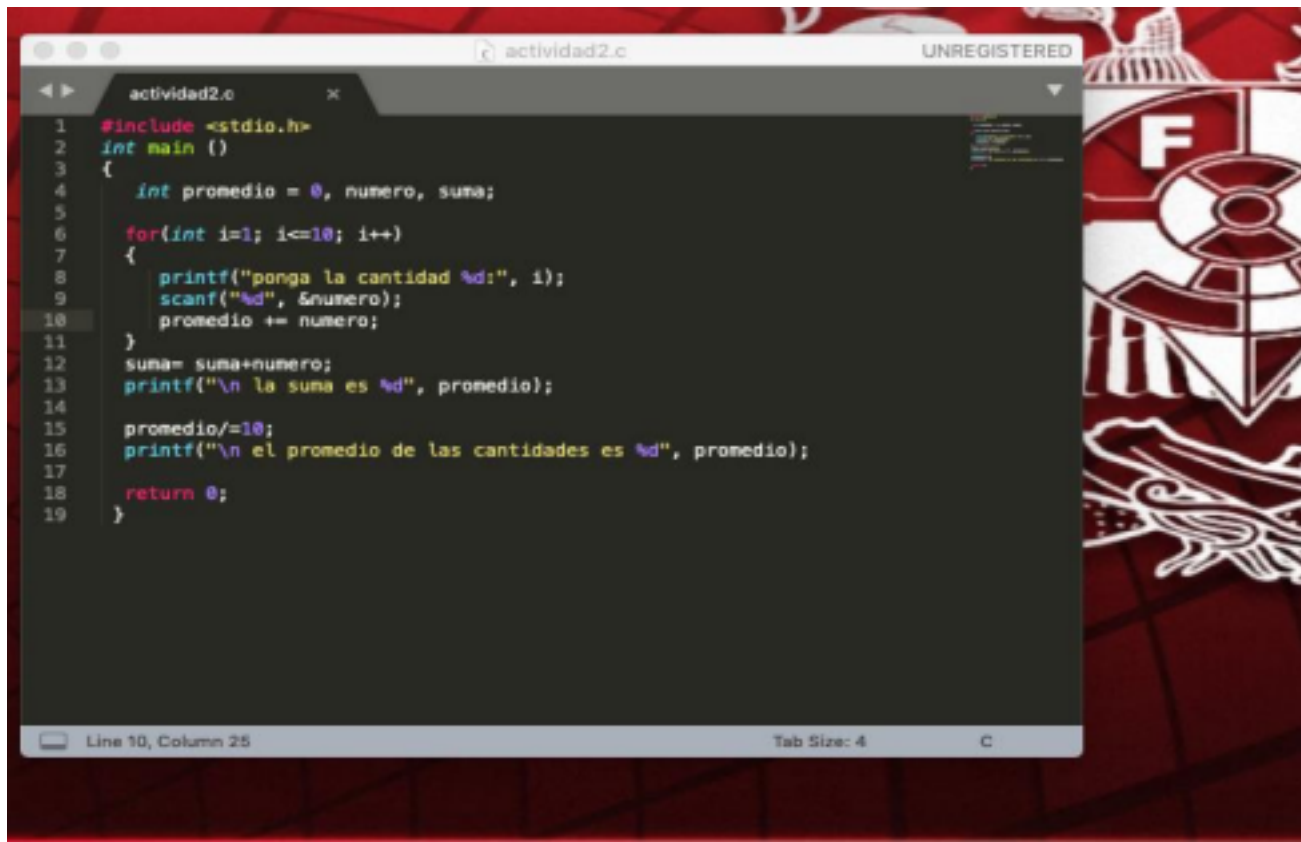
la tabla de multiplicar del 52 es:

1 * 52 = 52
2 * 52 = 104
3 * 52 = 156
4 * 52 = 208
5 * 52 = 260
6 * 52 = 312
7 * 52 = 364
8 * 52 = 416
9 * 52 = 468
10 * 52 = 520

desea ver otra tabla (s/n)?:
introduzca un numero entero: 
```

Actividad 2:

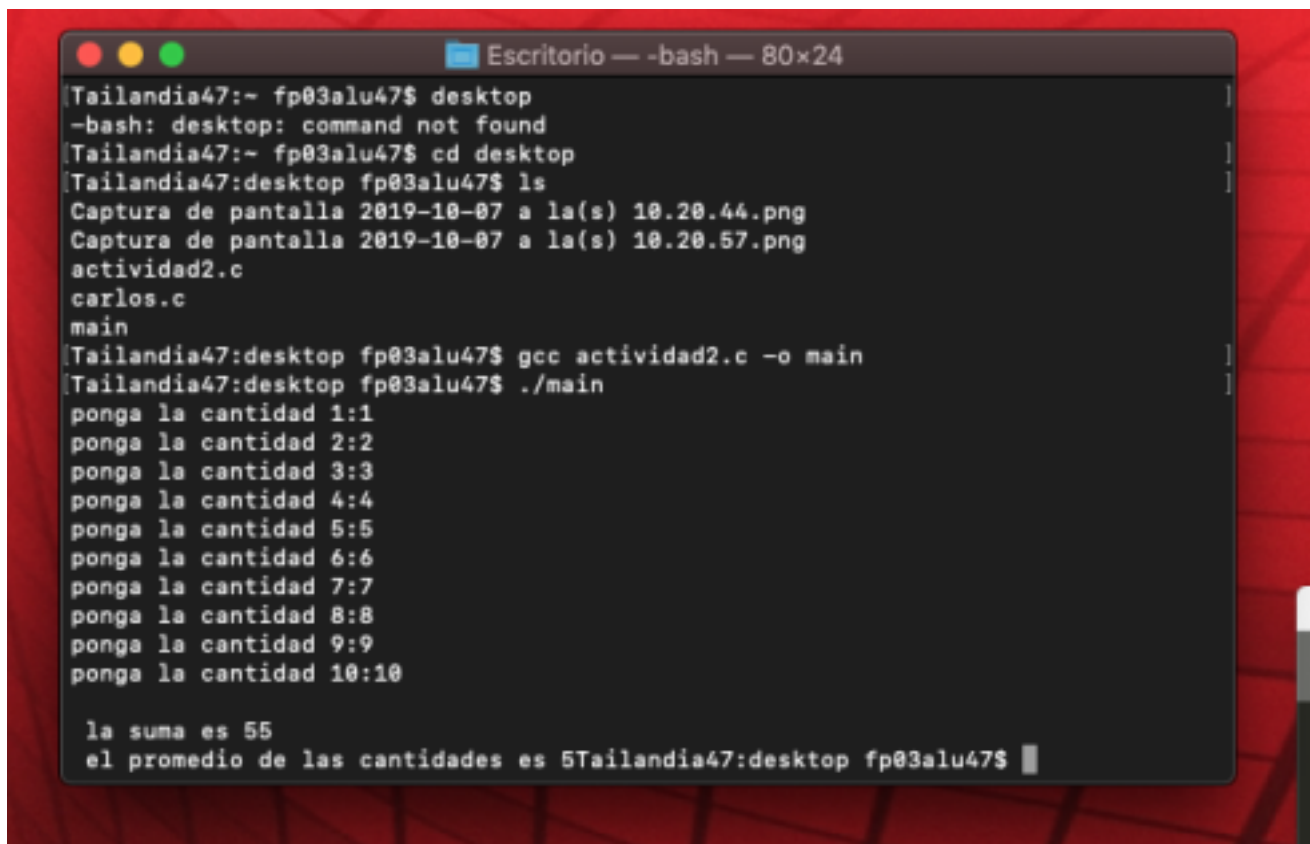
Hacer un programa que pida y lea 10 números y muestre su suma y su promedio.



The screenshot shows a code editor window titled 'actividad2.c' with a 'UNREGISTERED' watermark. The code is as follows:

```
1 #include <stdio.h>
2 int main ()
3 {
4     int promedio = 0, numero, suma;
5
6     for(int i=1; i<=10; i++)
7     {
8         printf("ponga la cantidad %d:", i);
9         scanf("%d", &numero);
10        promedio += numero;
11    }
12    suma= suma+numero;
13    printf("\n la suma es %d", promedio);
14
15    promedio/=10;
16    printf("\n el promedio de las cantidades es %d", promedio);
17
18    return 0;
19 }
```

The status bar at the bottom indicates 'Line 10, Column 25', 'Tab Size: 4', and 'C'.



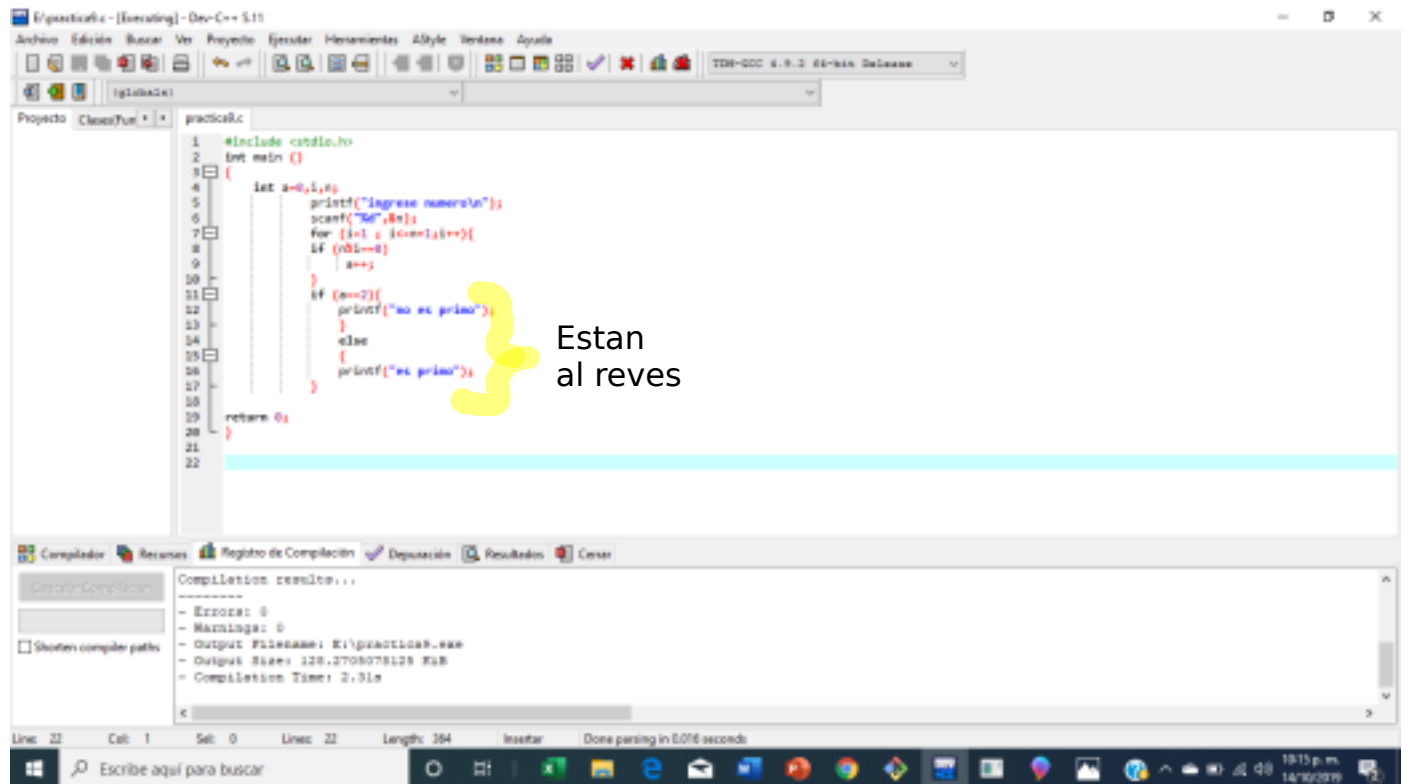
The screenshot shows a terminal window titled 'Escritorio — -bash — 80x24'. The user is in the directory 'fp03alu47\$' and has navigated to the 'desktop' directory. They have listed files, including 'actividad2.c', and compiled it with 'gcc actividad2.c -o main'. They then ran './main', which prompted them to enter 10 numbers (1 through 10). The program output shows the sum is 55 and the average is 5.

```
Tailandia47:~ fp03alu47$ desktop
-bash: desktop: command not found
Tailandia47:~ fp03alu47$ cd desktop
Tailandia47:desktop fp03alu47$ ls
Captura de pantalla 2019-10-07 a la(s) 10.20.44.png
Captura de pantalla 2019-10-07 a la(s) 10.20.57.png
actividad2.c
carlos.c
main
Tailandia47:desktop fp03alu47$ gcc actividad2.c -o main
Tailandia47:desktop fp03alu47$ ./main
ponga la cantidad 1:1
ponga la cantidad 2:2
ponga la cantidad 3:3
ponga la cantidad 4:4
ponga la cantidad 5:5
ponga la cantidad 6:6
ponga la cantidad 7:7
ponga la cantidad 8:8
ponga la cantidad 9:9
ponga la cantidad 10:10

la suma es 55
el promedio de las cantidades es 5Tailandia47:desktop fp03alu47$
```

Actividad 3:

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

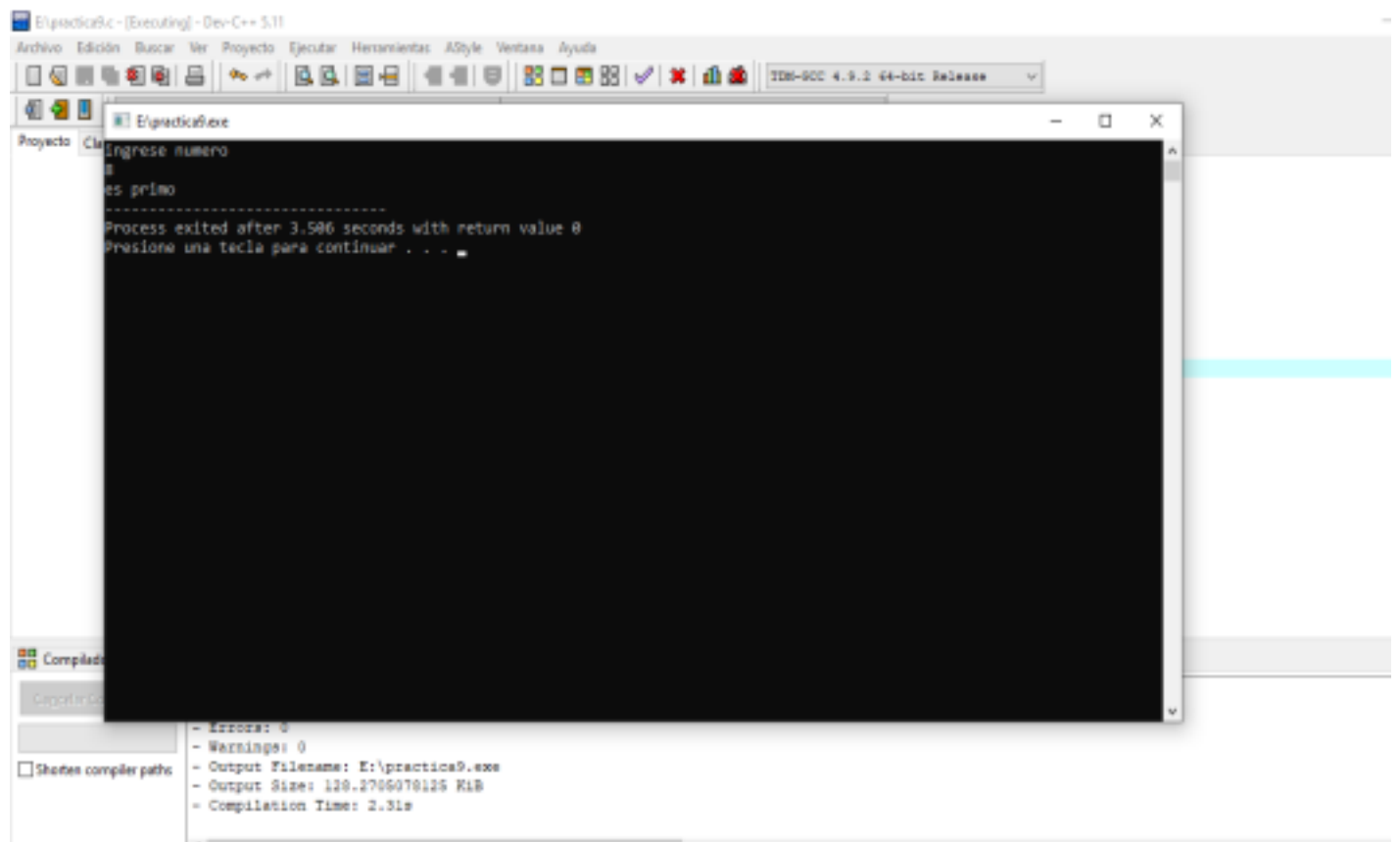


```
1 #include <stdio.h>
2 int main ()
3 {
4     int a=0,i,e;
5     printf("ingrese numero\n");
6     scanf("%d",&a);
7     for (i=1; i<=a; i++){
8         if (a%i==0)
9             a++;
10    }
11    if (a==2){
12        printf("no es primo");
13    }
14    else
15    {
16        printf("es primo");
17    }
18    return 0;
19 }
```

Estan al revés

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: E:\practicas9.exe
- Output Size: 128.2708078125 KkB
- Compilation Time: 2.31s



```
E:\practica9.exe
ingrese numero
8
es primo
-----
Process exited after 3.586 seconds with return value 0
Presione una tecla para continuar . . .
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: E:\practica9.exe
- Output Size: 128.2708078125 KkB
- Compilation Time: 2.31s

REPORTE DE PRÁCTICA:

no es verdad

En esta practica vimos las expresiones lógicas más básicas que hay en el lenguaje de tipo C, con base a ellas nos pusimos a hacer estas actividades, con lo que anteriormente nosotros ya sabíamos con las clases de teoria y todos los ejercicios que hemos hecho en clase ya que todos ellos nos han servido para llevar a cabo este tipo de programas, ya que Podemos compilarlos y correrlos, Podemos ver en que estemos mal, ya que en un ejercicio escrito se complica porque no sabes en que estarás mal o en que no, aun esto se pone mas interesante, cada vez esta subiendo más y más en nivel de dificultad.