|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **NOMBRE DE LA PRÁCTICA** | **MANUAL DE PRACTICAS UNIDAD 1** | | | **No. DE PRACTICA** | **1** |
| **ASIGNATURA:** | **METODOS NUMERICOS** | **CARRERA:** | **INGENIERÍA EN SISTEMAS COMPUTACIONALES** | **PLAN:** | **ISIC-2010-224** |

**NOMBRE: MARTIN SANCHEZ MERA GRUPO: 3402**

No. Cuenta 14230112

1. **COMPETENCIA(S) ESPECÍFICA(S):**

Aplica el paradigma orientado a objetos para el desarrollo de aplicaciones que solucionen problemas de su entorno.

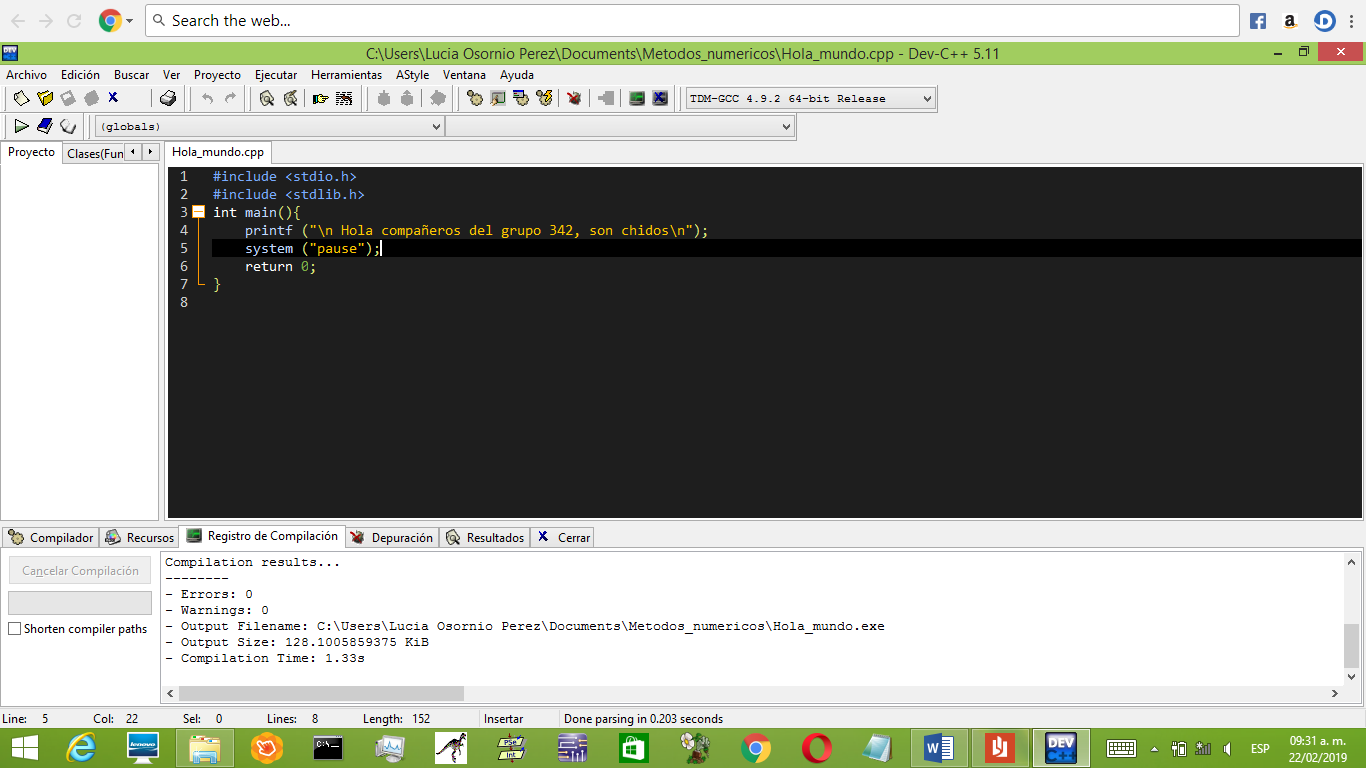
**MATERIAL EMPLEADO:**

* WORD
* DEV C++

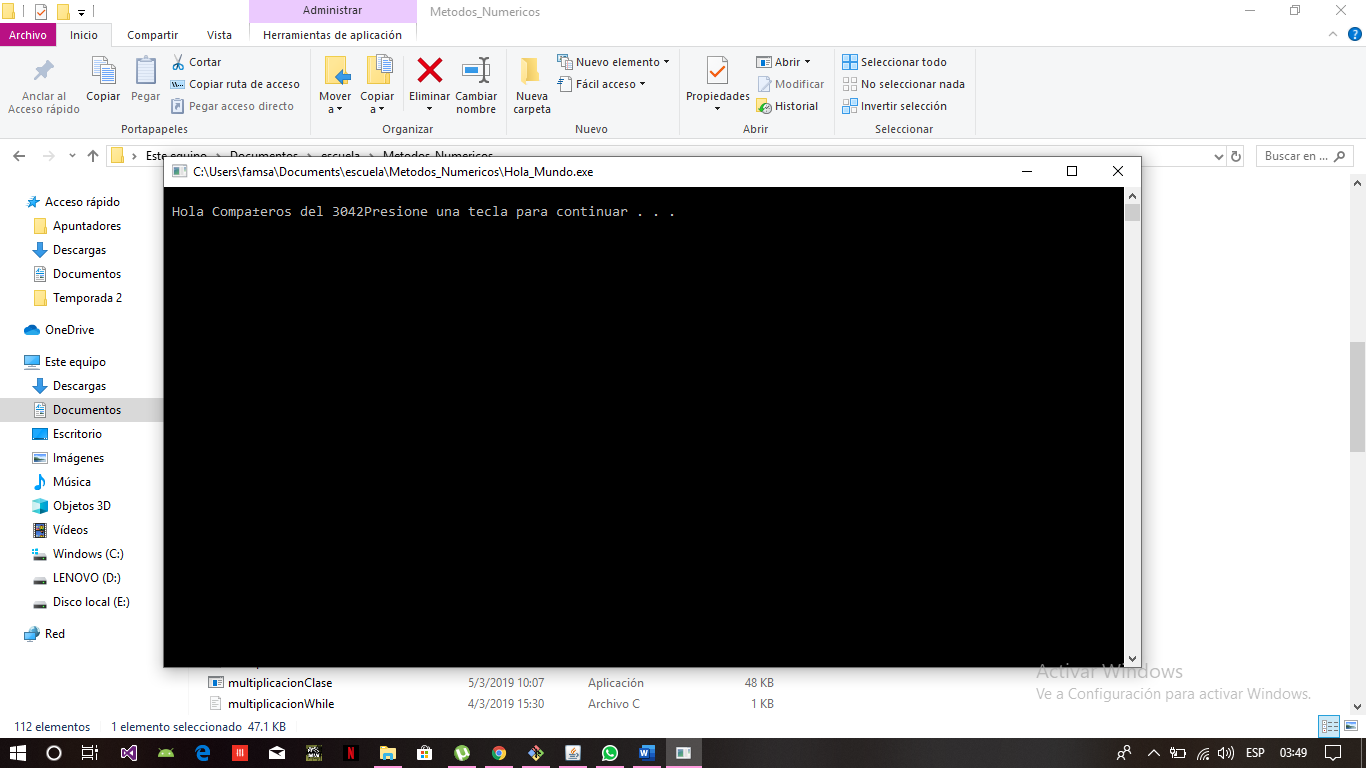
1. **DESARROLLO DE LA PRÁCTICA:**

* Para empezar a realizar los programas para mostrar los mensajes en pantalla se utiliza la función **printf**
* Para poder usar la función debemos incluir en el proyecto la Biblioteca **stdio.h**.
* Se utiliza el carácter especial **\n** para realizar un salto de línea.

**Lenguaje C**

Diseñamos nuestro primer programa en C++ con un mensaje de hola mundo

Ya una vez guardado se compila y aparece el mensaje



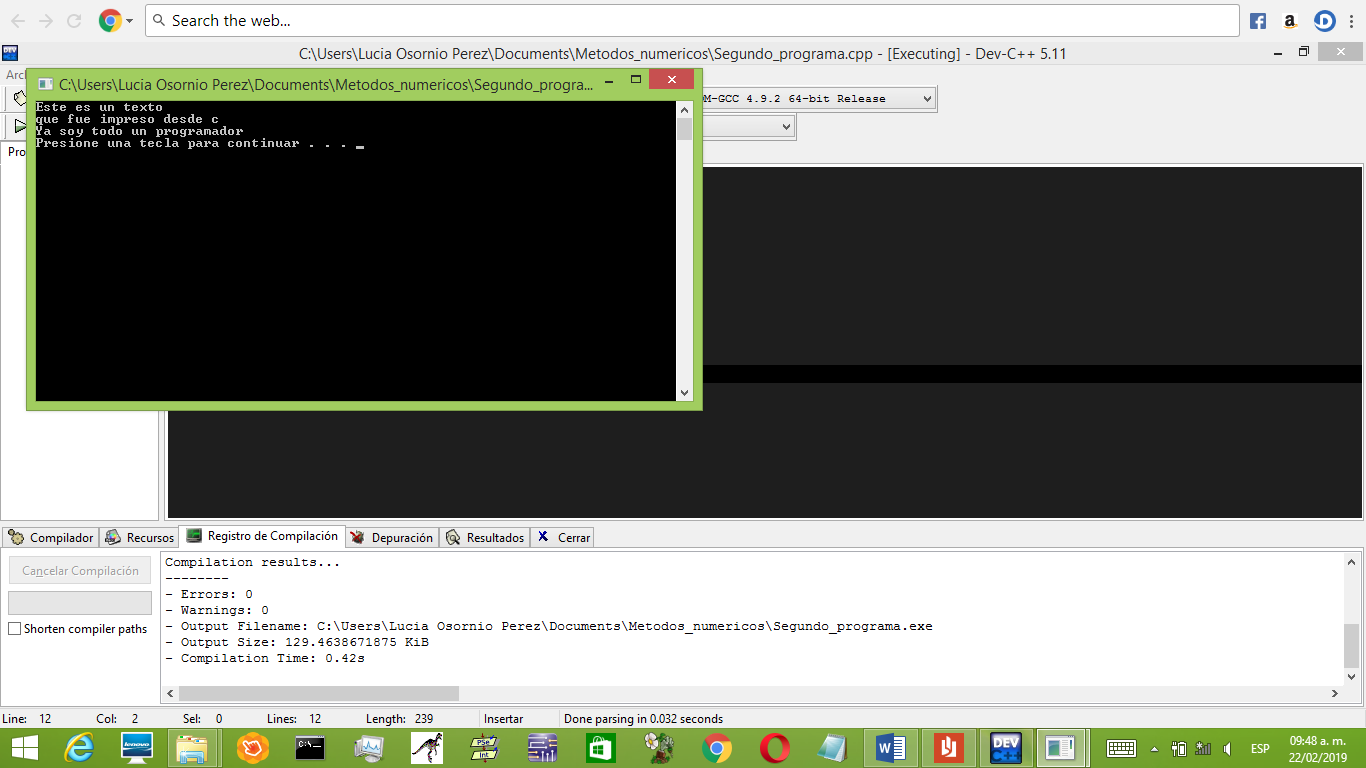
Estos son algunos caracteres espéciales que no son imprimibles en el programa:

\n Salto de línea 

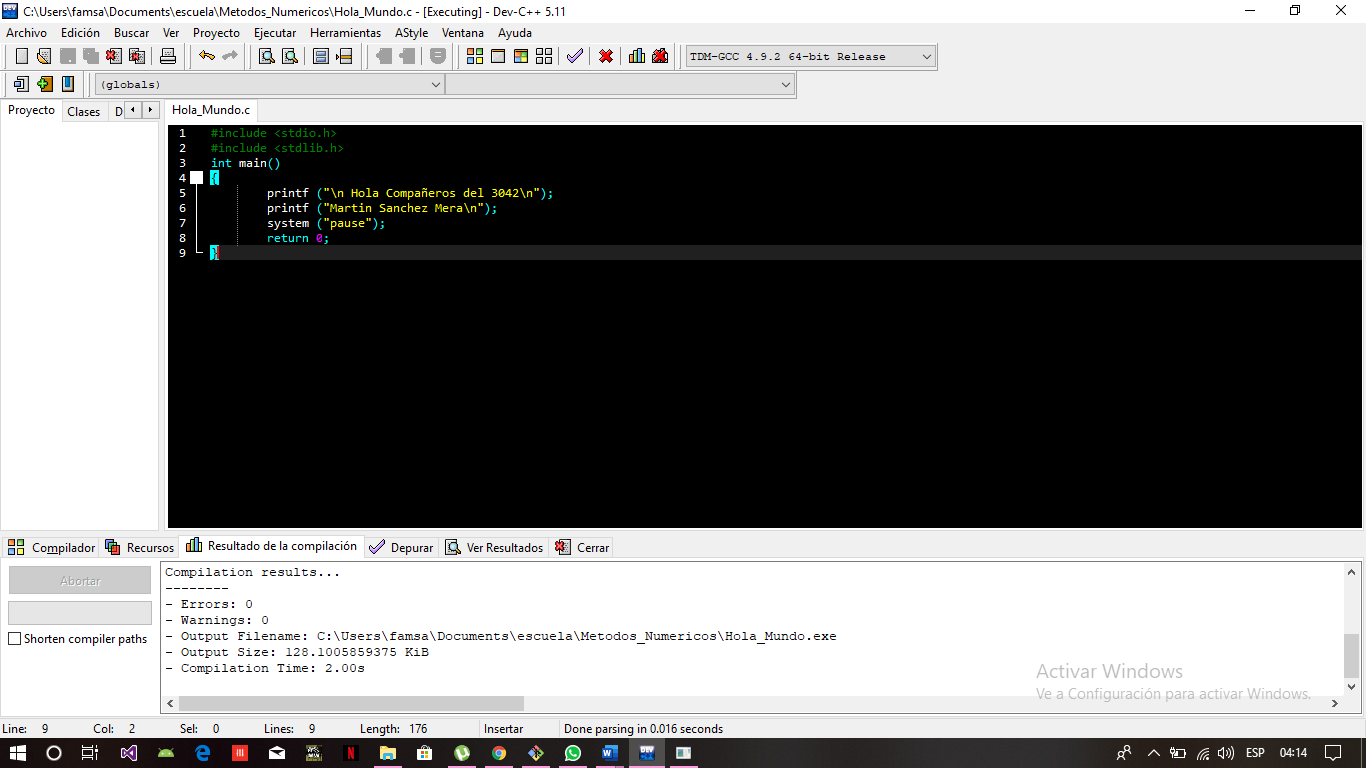
\t Imprime un espacio de tabulador horizontal 

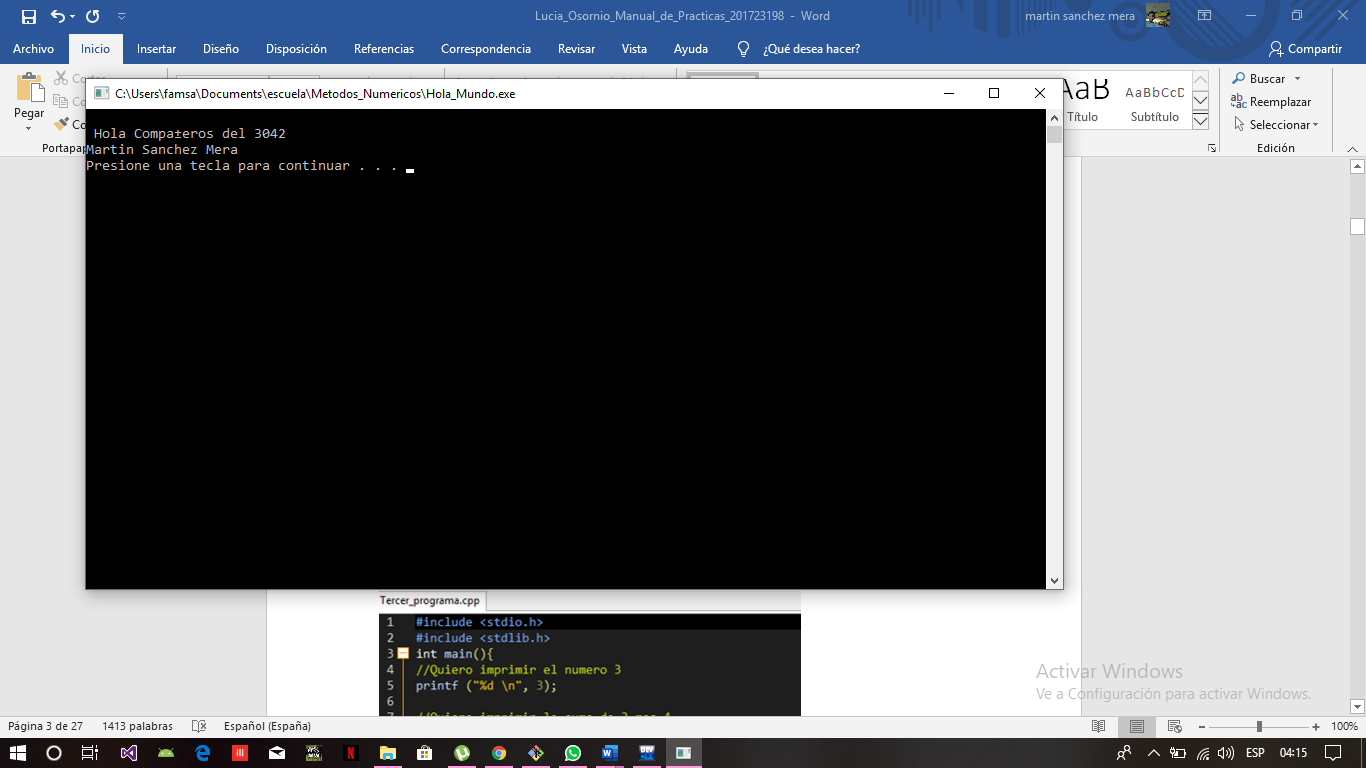
\a Sonido audible 

\b Retroceso



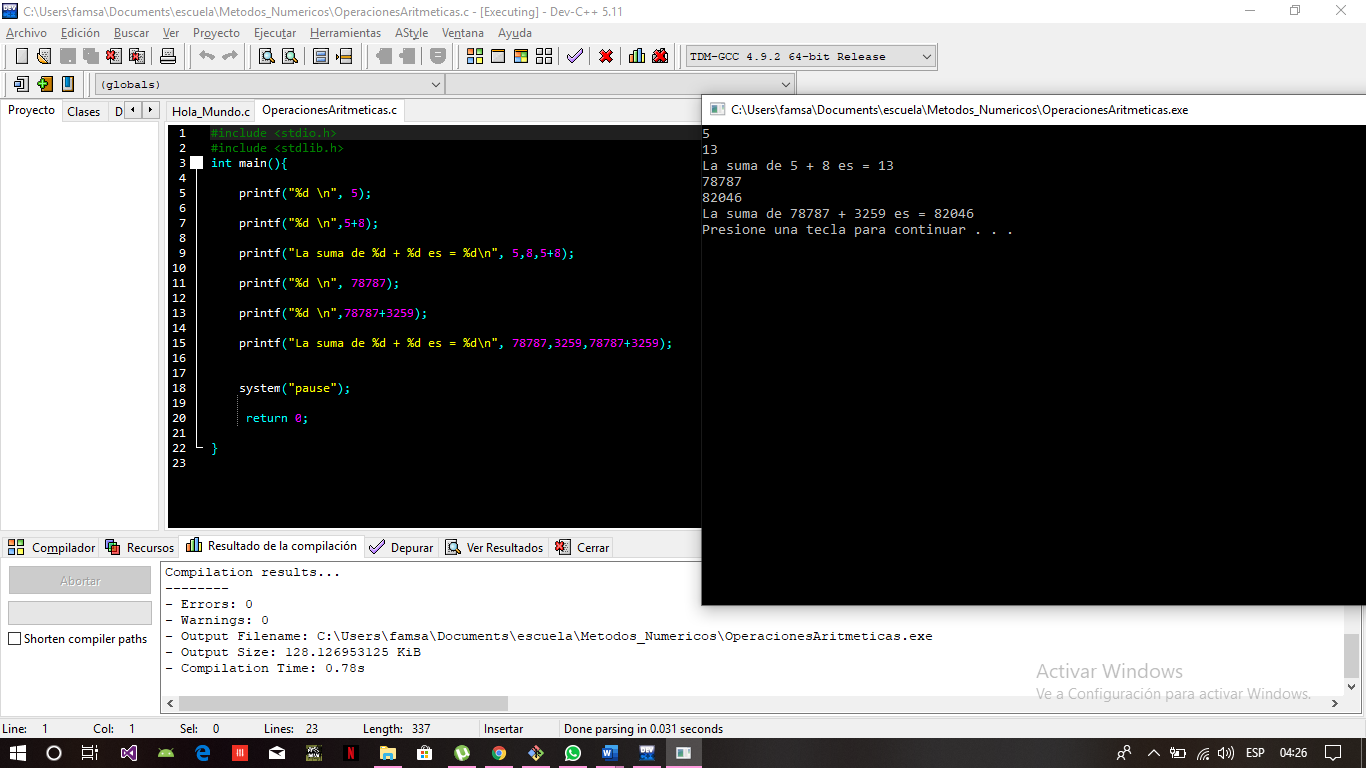
El siguiente programa fue una bienvenida como ejercicio del tema anterior





**Operaciones matemáticas**

Se hará dos sumas de números enteros y podemos notar que **%d** se utiliza para cualquier tipo de dato entero.



**Operadores aritméticos**

+ [Adición] % [módulo]

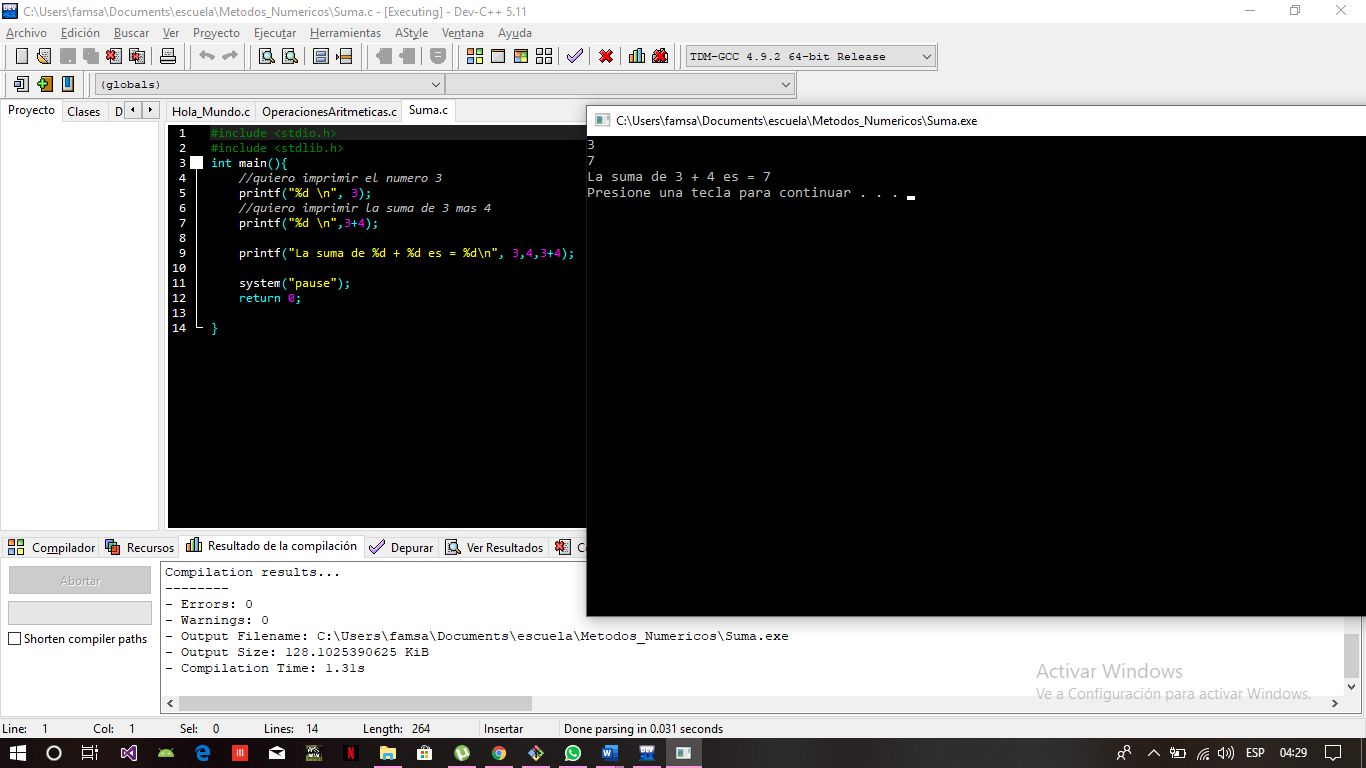
++ [Incremento] -- [Decremento]

- [sustracción] \* [Multiplicación] / [División

Las palabras reservadas son identificadores utilizados por el lenguaje para un fin específico. No pueden ser empleadas por el programador para nombrar variables, atributos, métodos o clases.

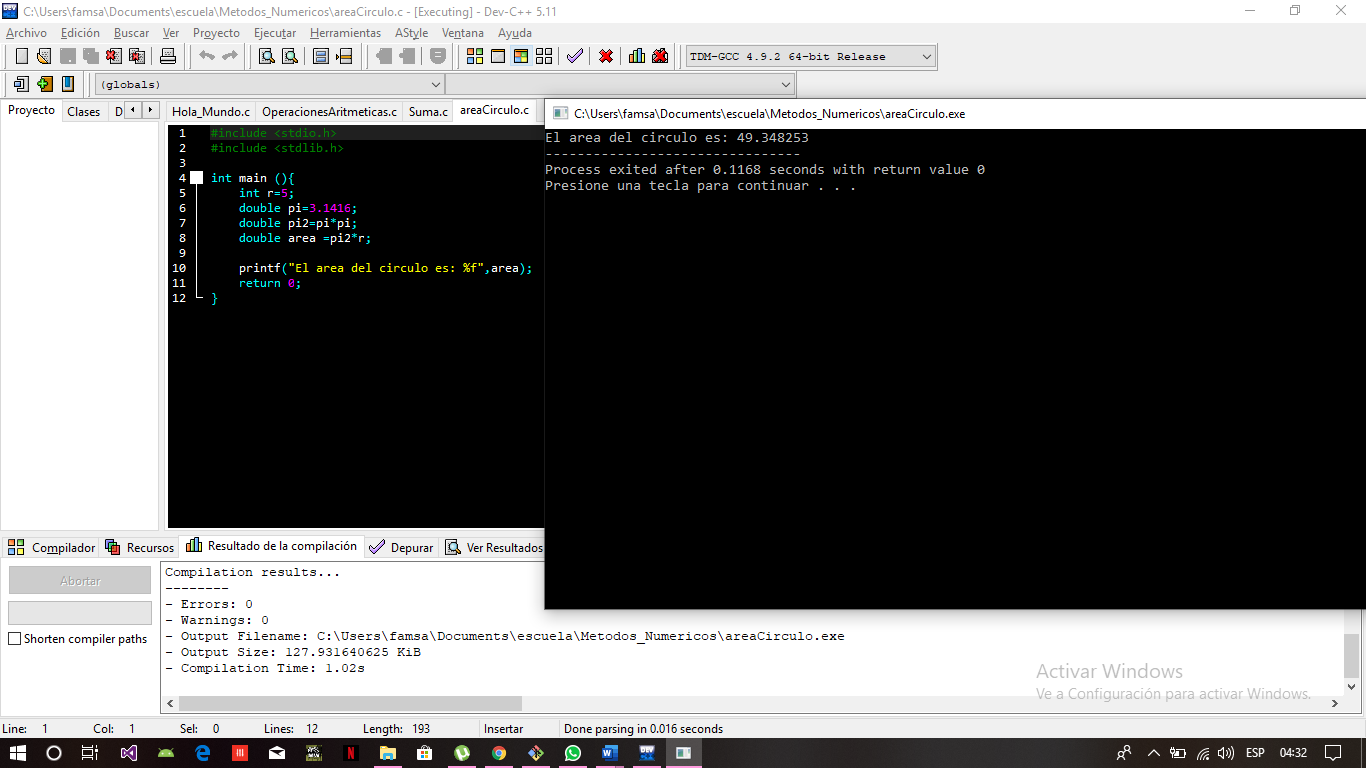
**Ejercicio:**

Realizamos operacipon %d (suma).

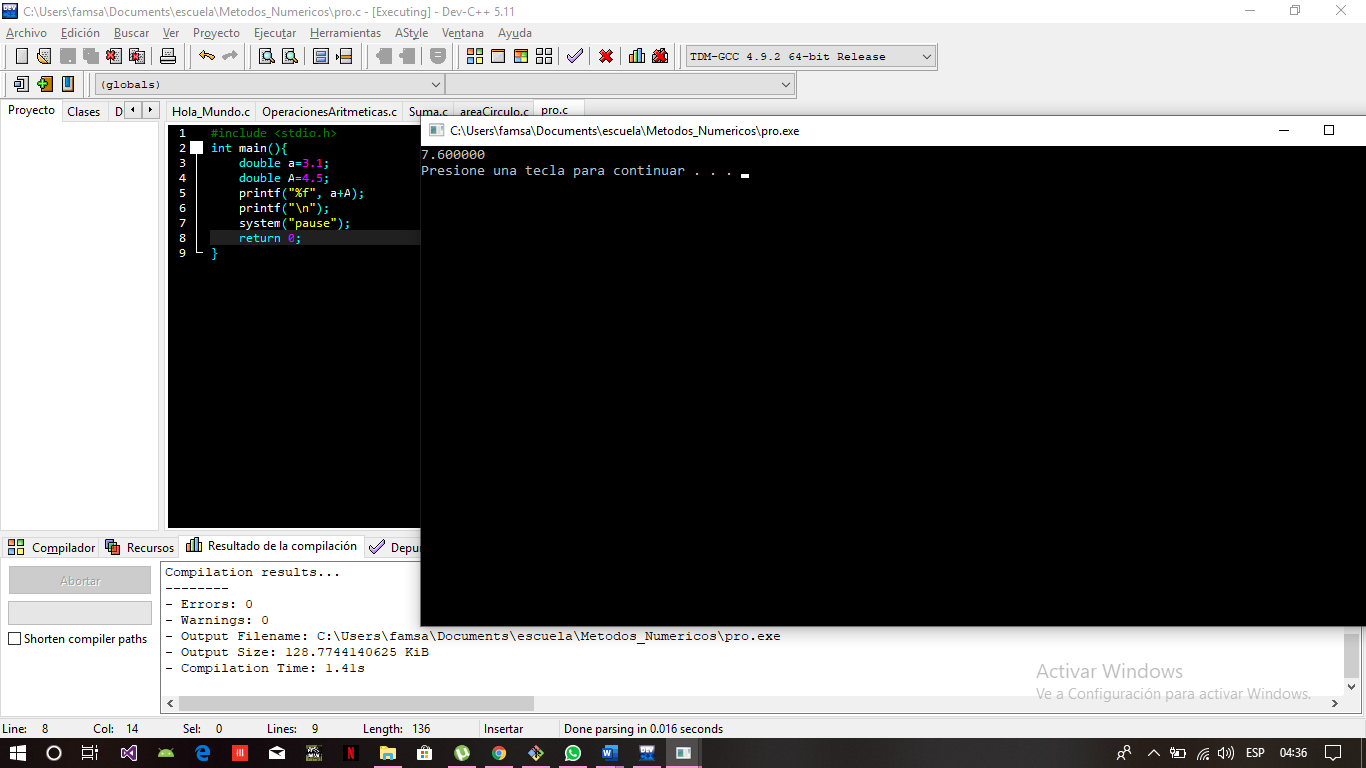


**Ejercicio:** ….

Declaramos la constante usando la palabra #**define** y mostrando que se puede realizar una impresión de números de punto flotante con **%f**.



Ejercicio



**Ejercicio:**

Calcula la cantidad de segundos que has vivido.

1 min = 60 seg.

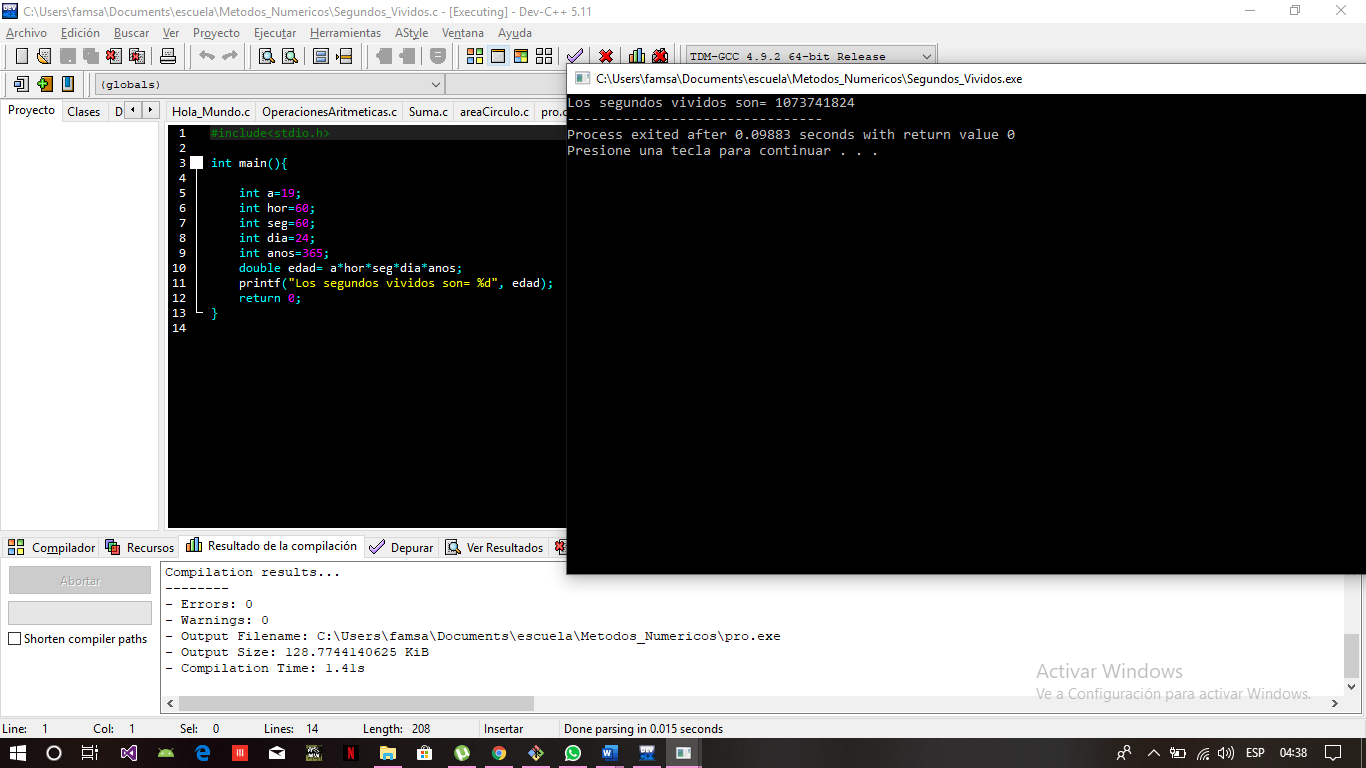
1 hora = 60 min.

1 día = 24 horas.

1 año = 365 días.

**Ejercicio:**

Declarar 3 variables ya signarles valores enteros. Mostrar el resultado de la multiplicación de las 3 variables.



**Ejercicio:**

Crea un programa que calcule el área de un círculo quetiene10metrosdediámetro.

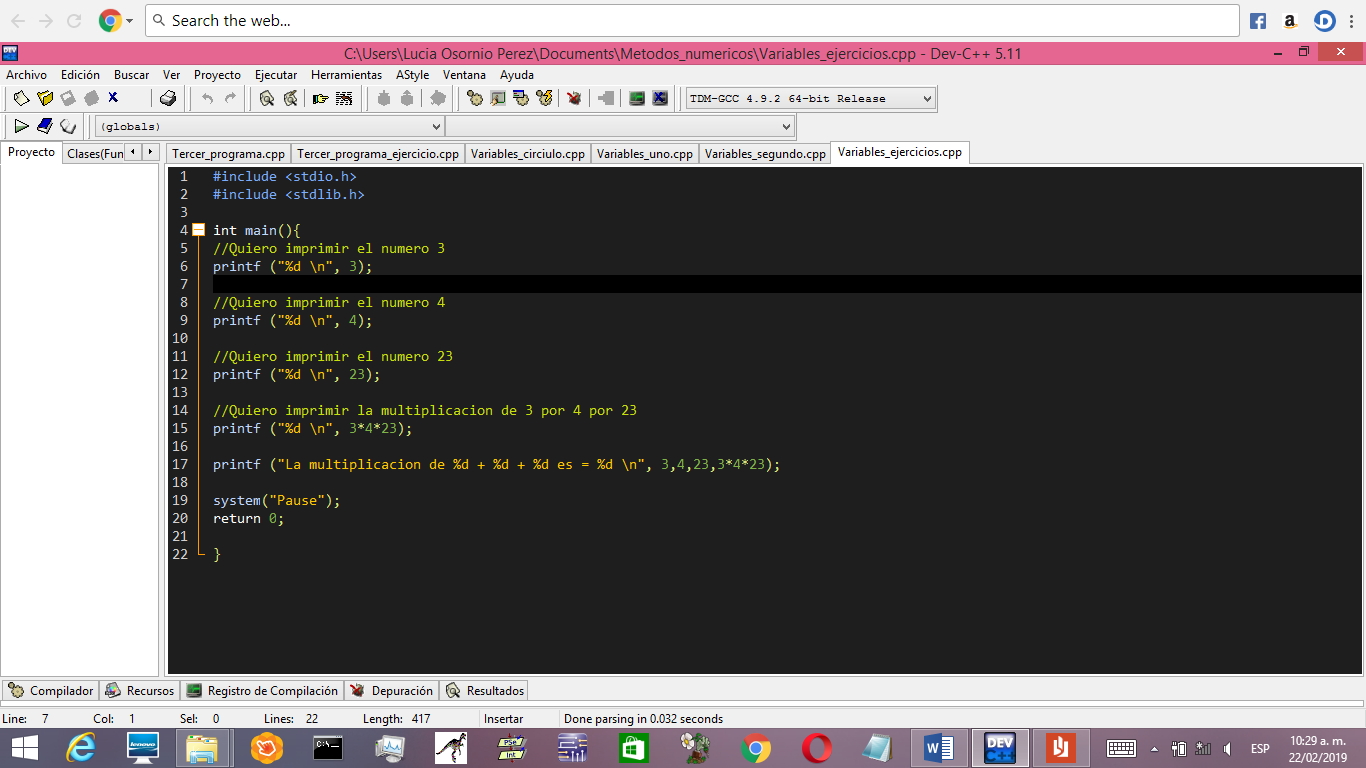
Área = πr2

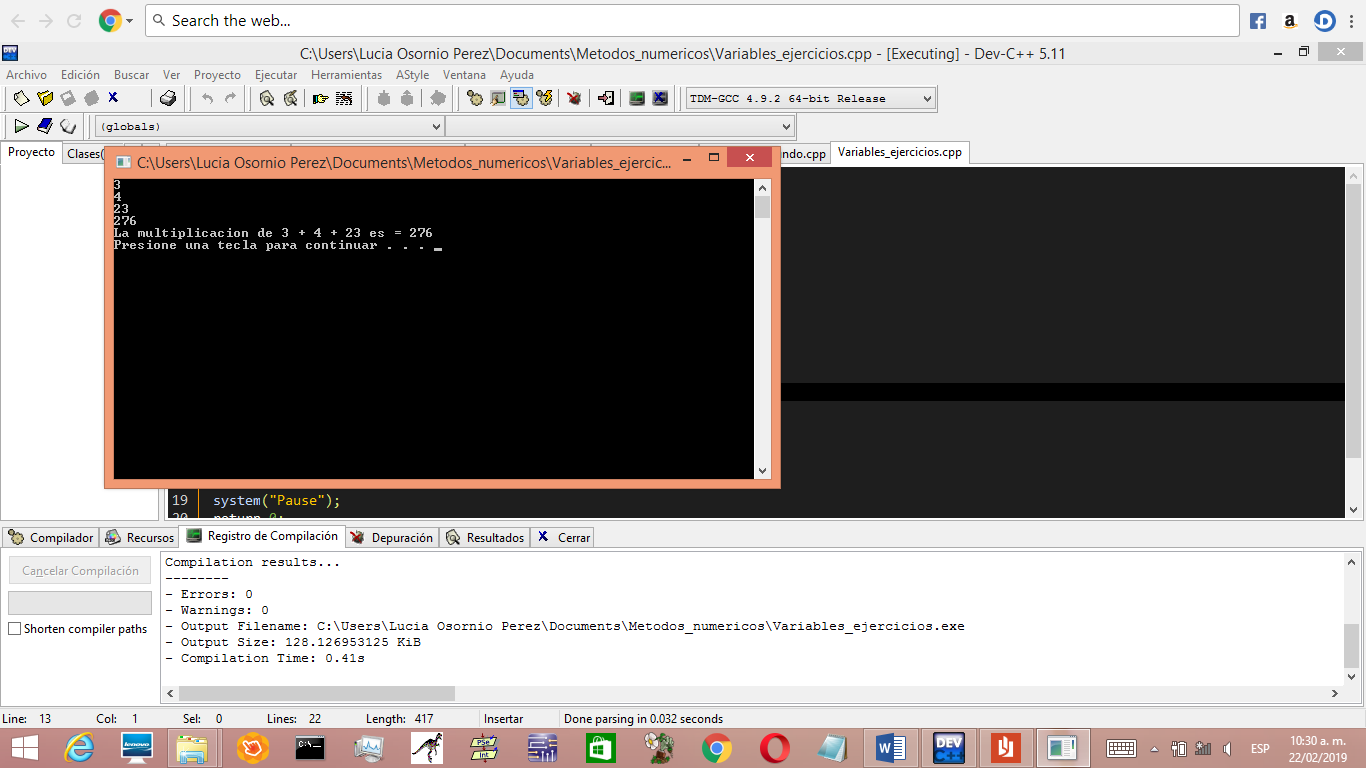
Aquí se muestran algunos ejercicios de los problemas planteados

Ejericio1 

Declarar 3variables y asignarles valores enteros. 

Mostrar el resultado de la multiplicación de las3 variables

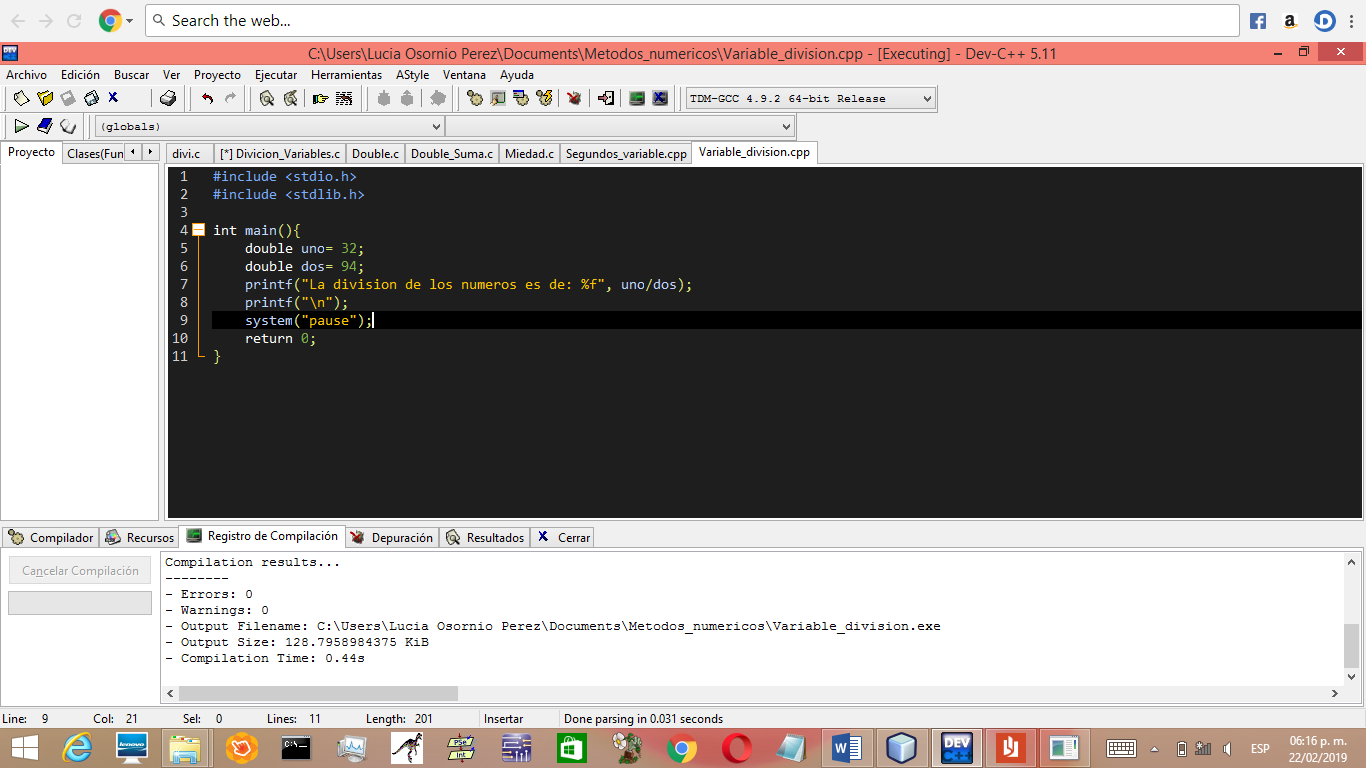


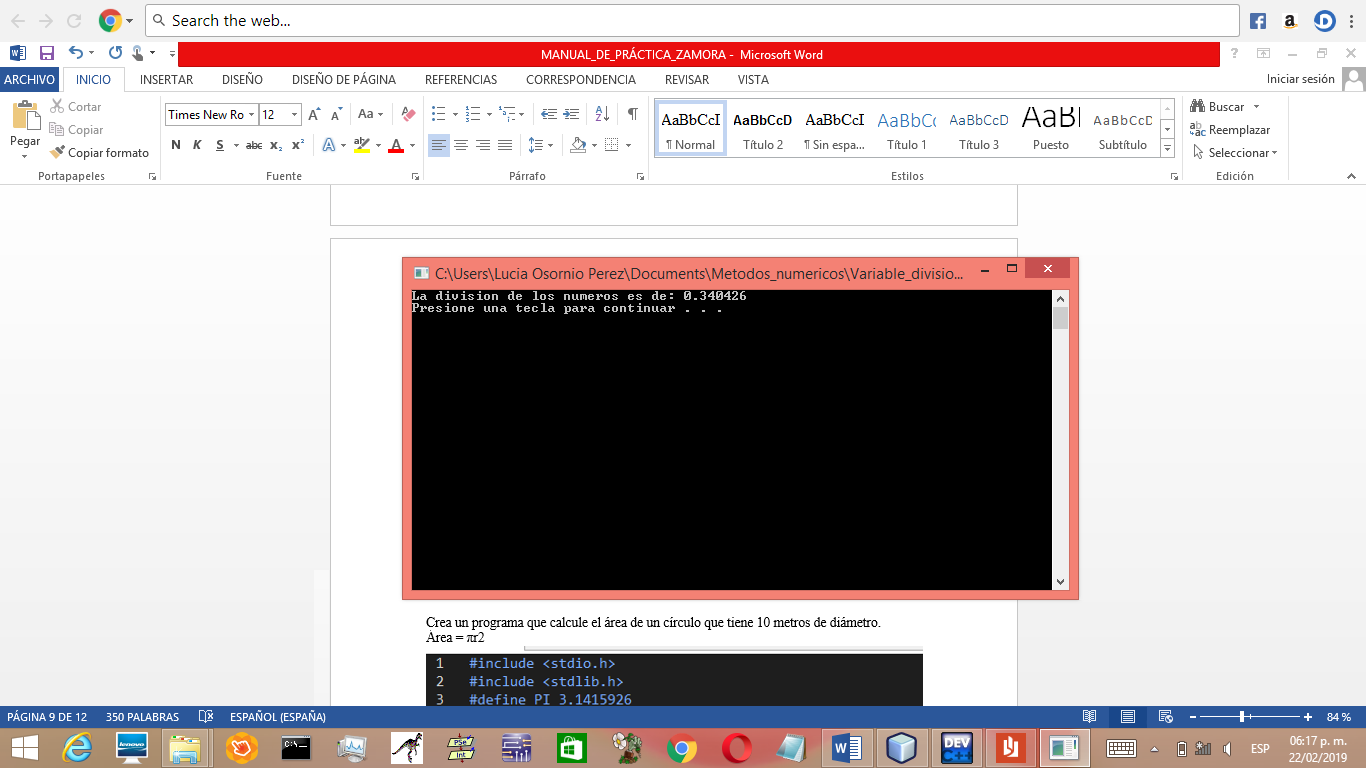


Ejercicio2 

Declarar 2 variables dobles (reales) y asignarles valores diferentes de 0.

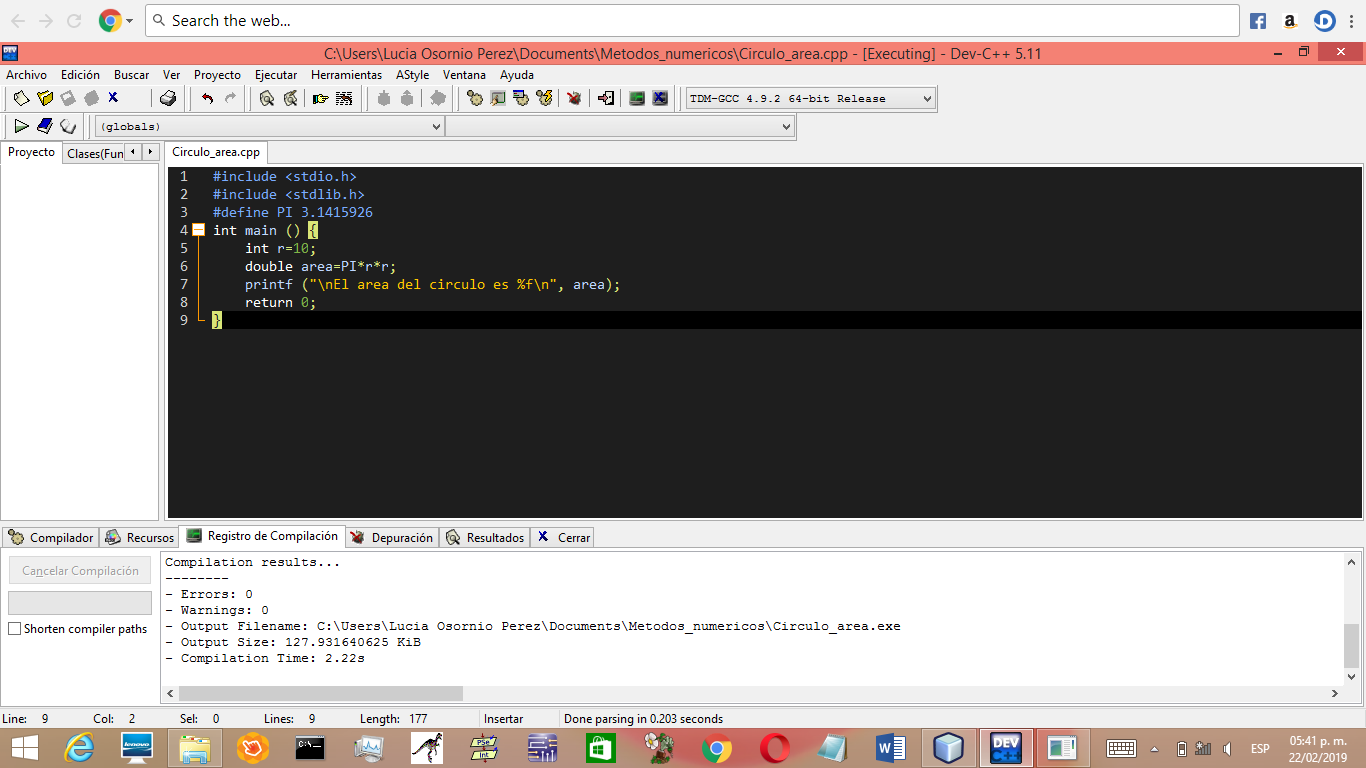
Mostrar el resultado de dividir la primera entre la segunda.

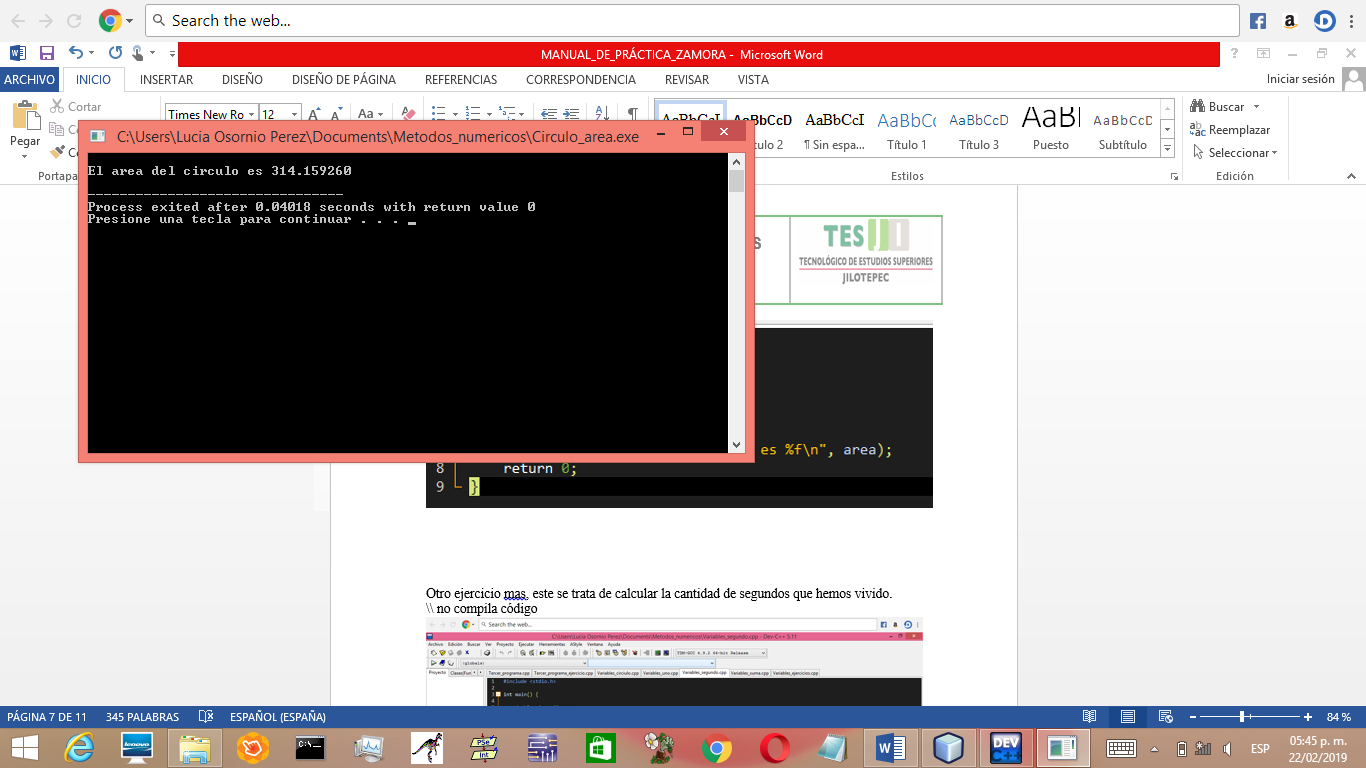




Crea un programa que calcule el área de un círculo que tiene 10 metros de diámetro.

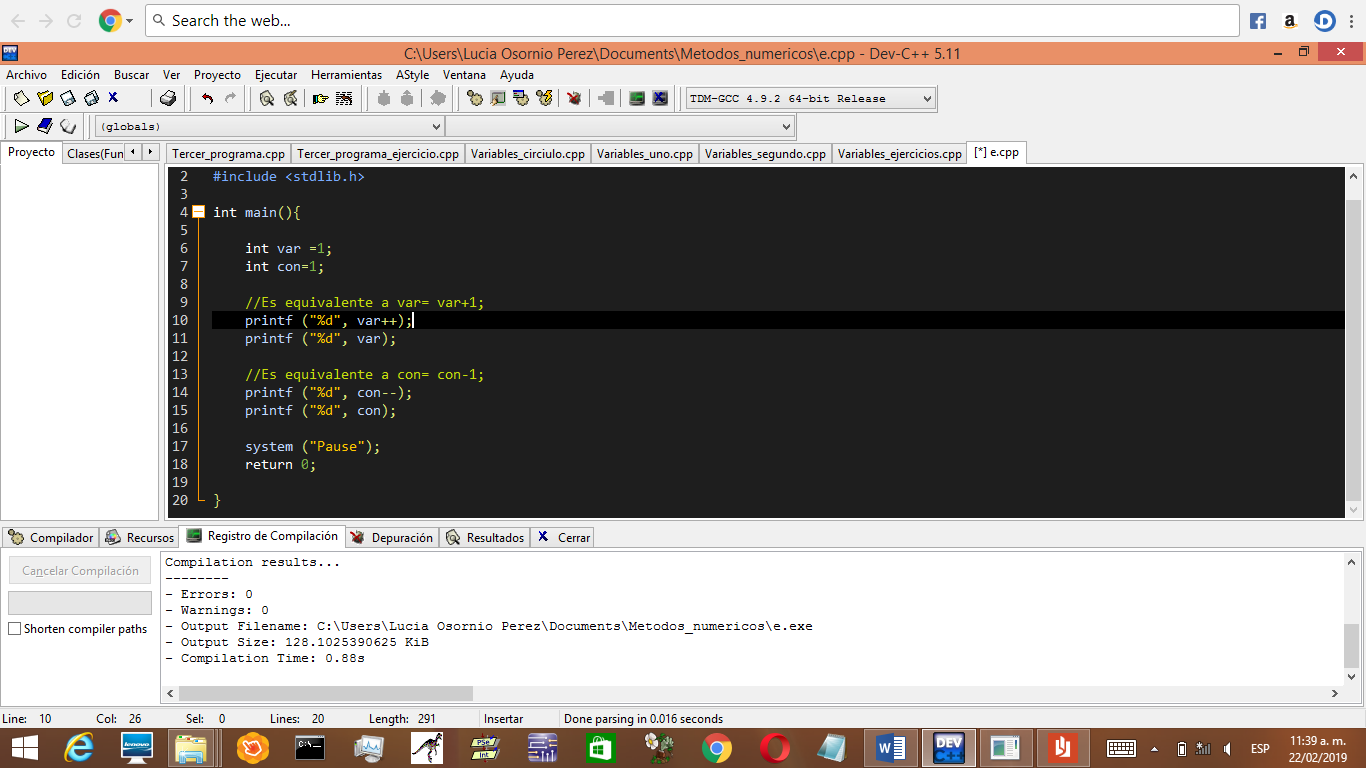
Área = πr2

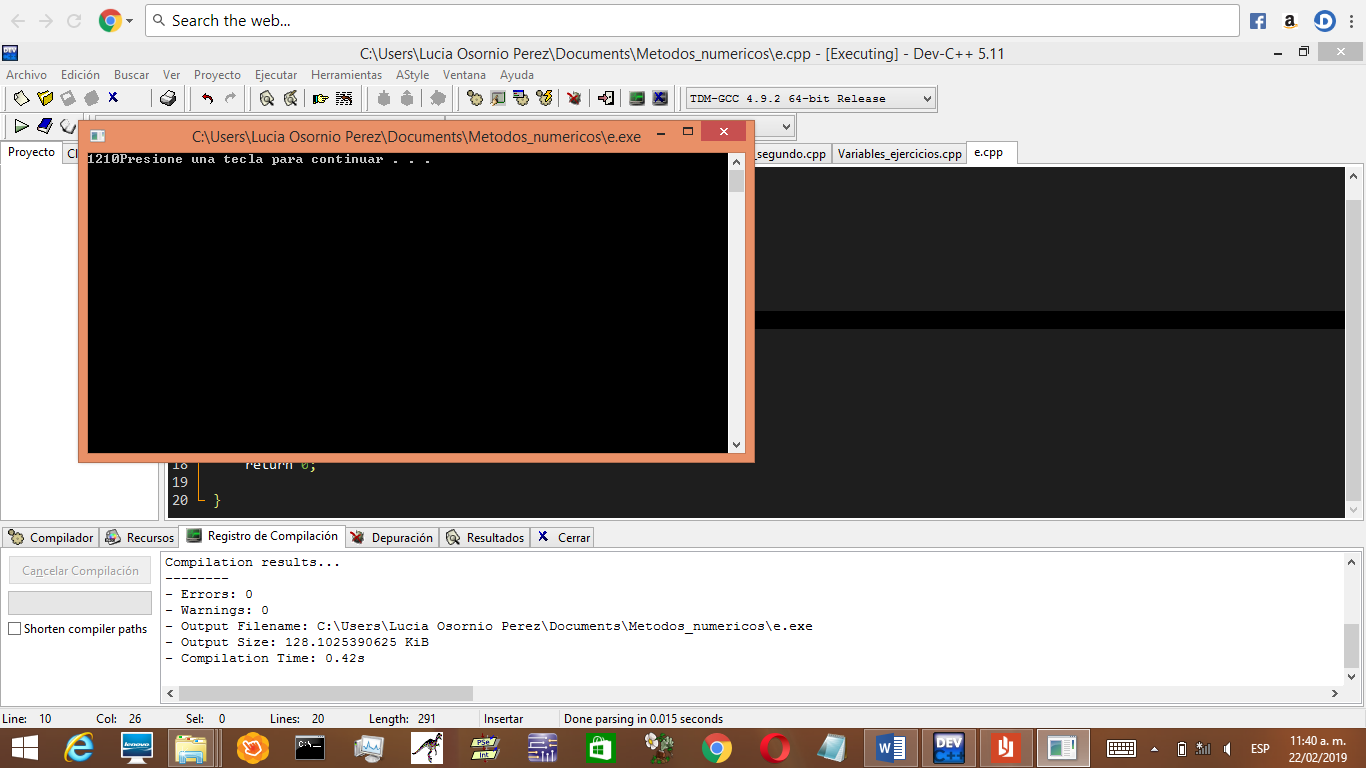




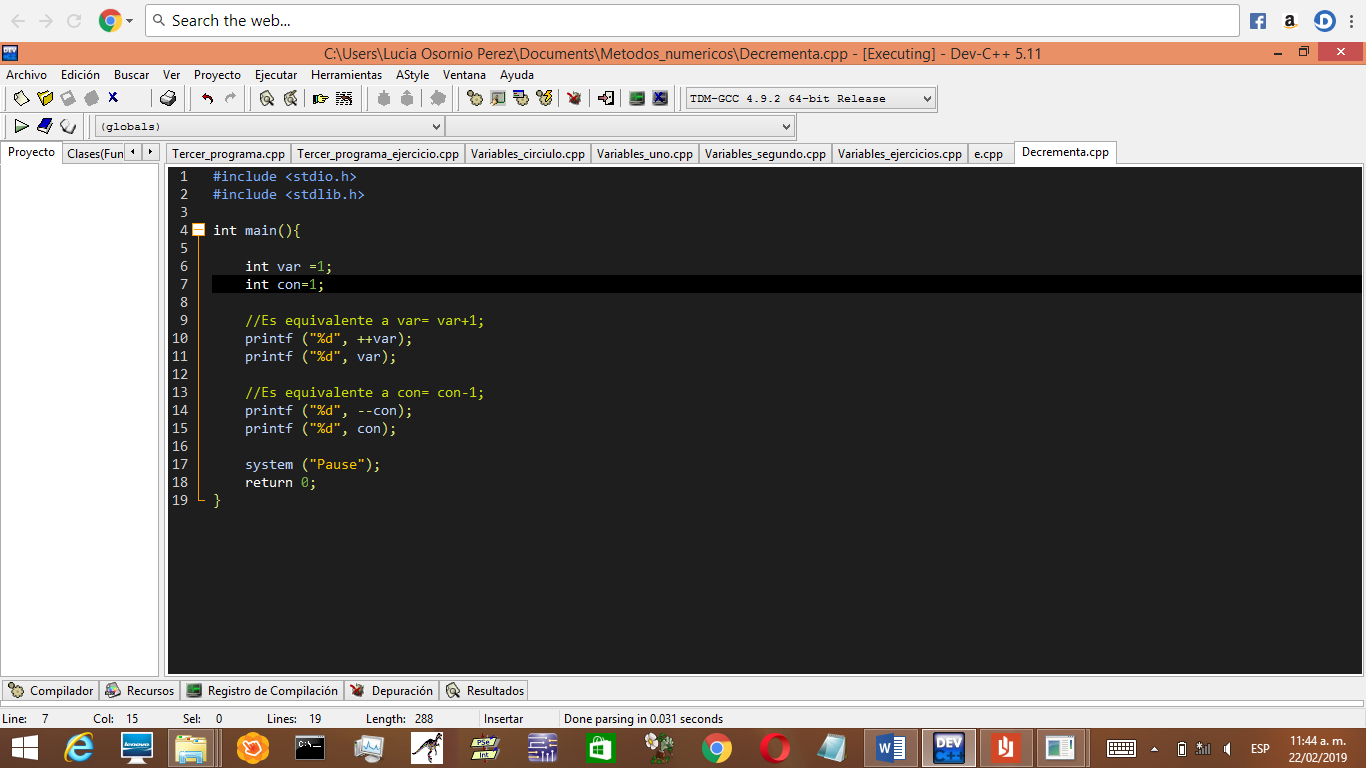
**Operador incremento\ decremento**

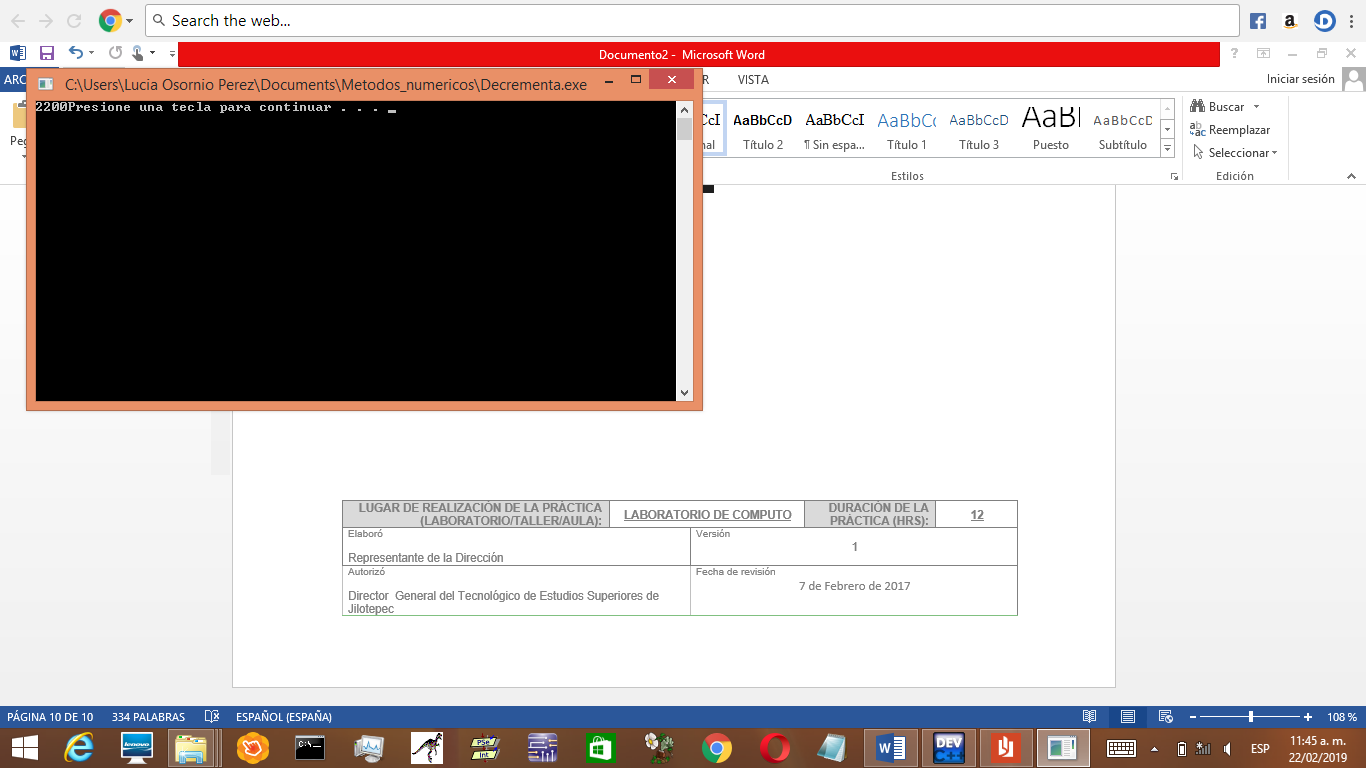
Crea un programa donde se ejecuta un post incremento

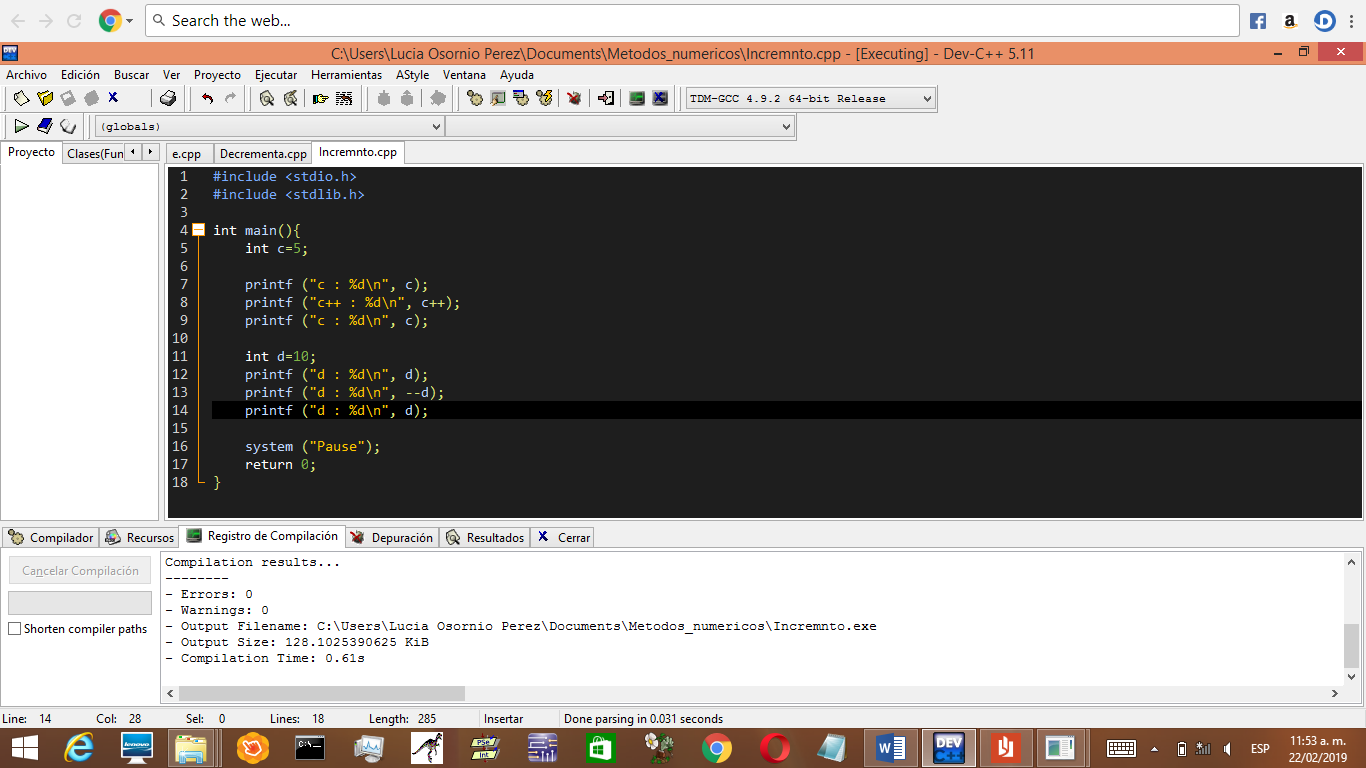


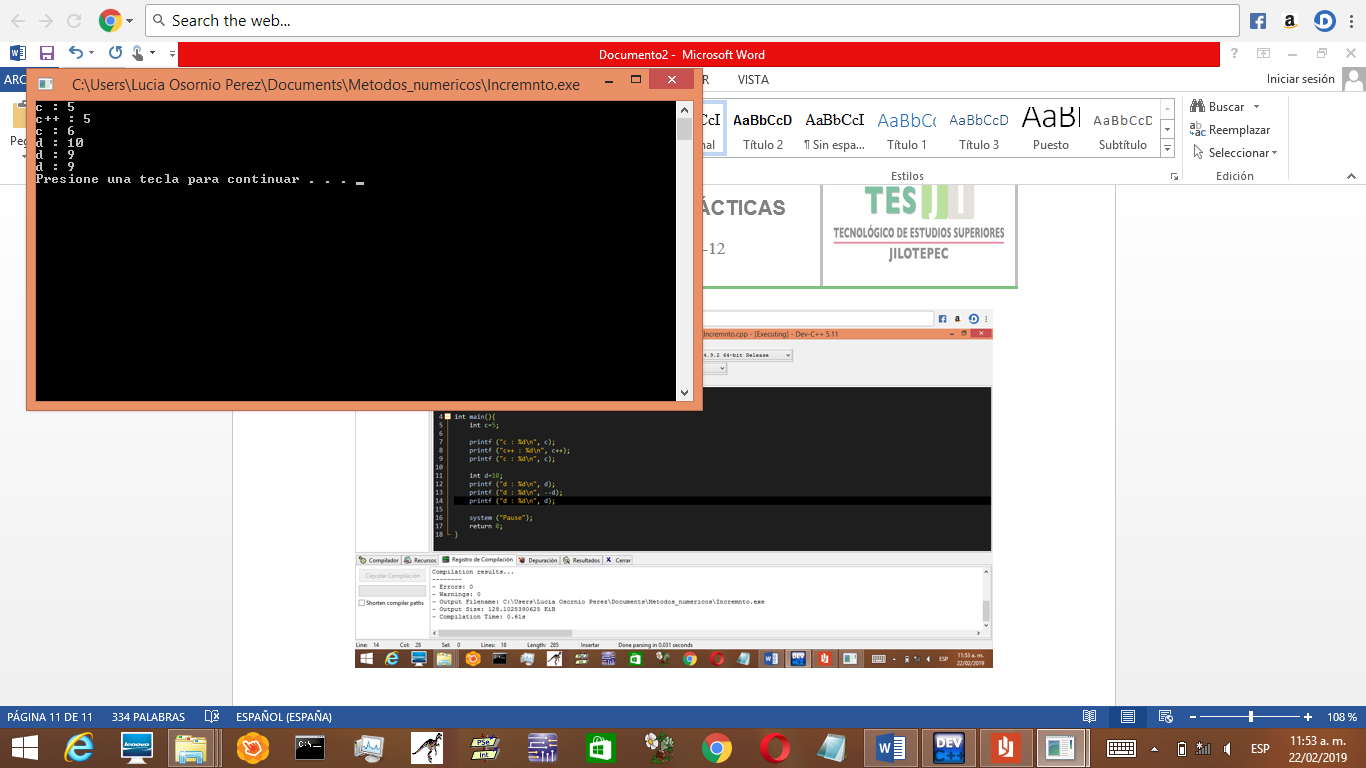


En este programa, la variable se decrementa

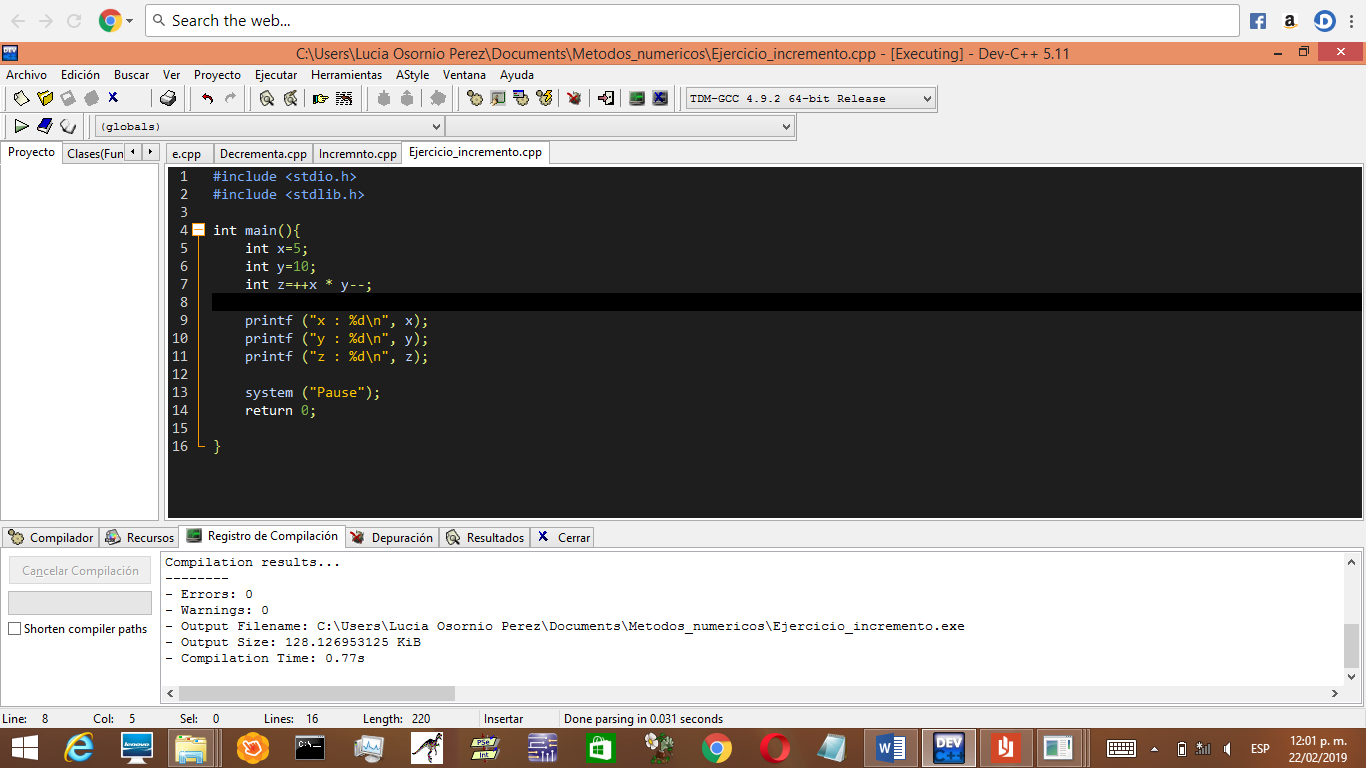


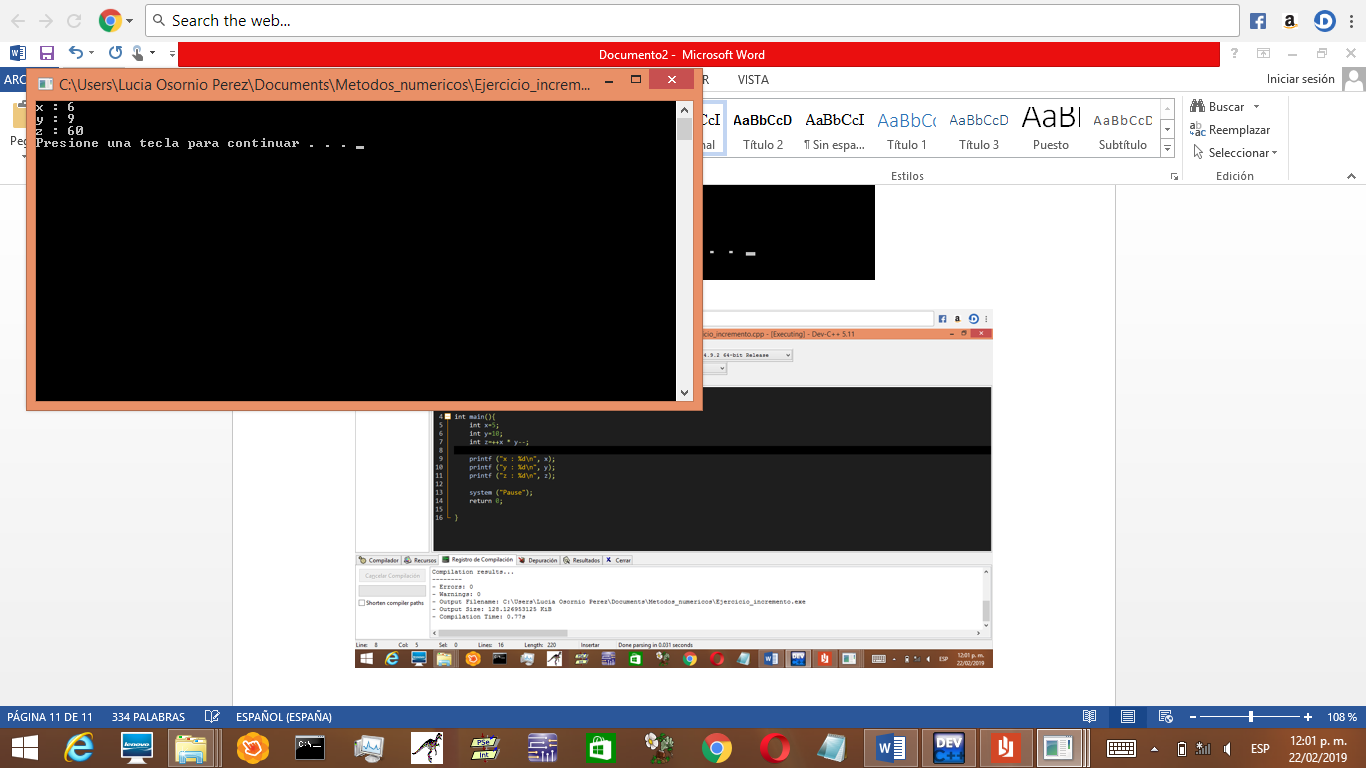




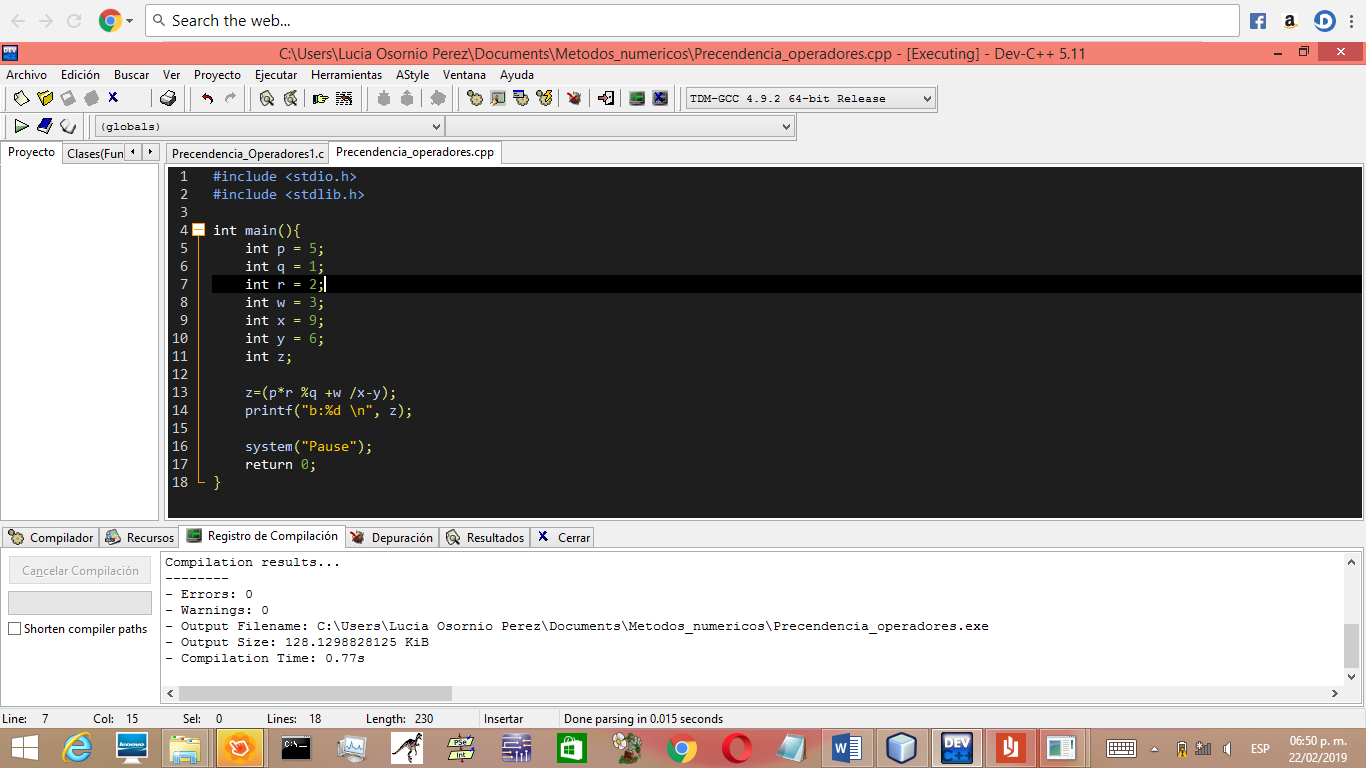


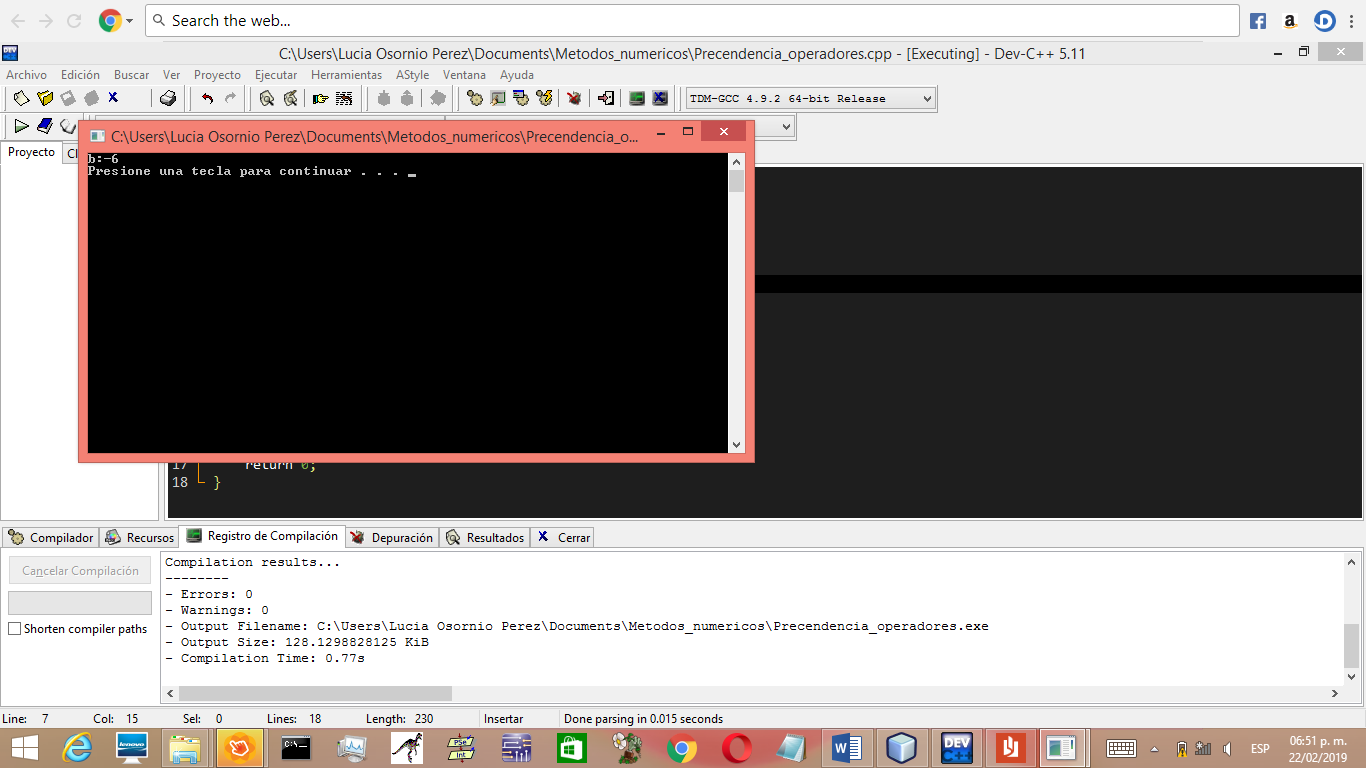
Este programa es un ejemplo mas de incremento, decremento…¿Cuánto vale Z?





**Precedencia de operadores**





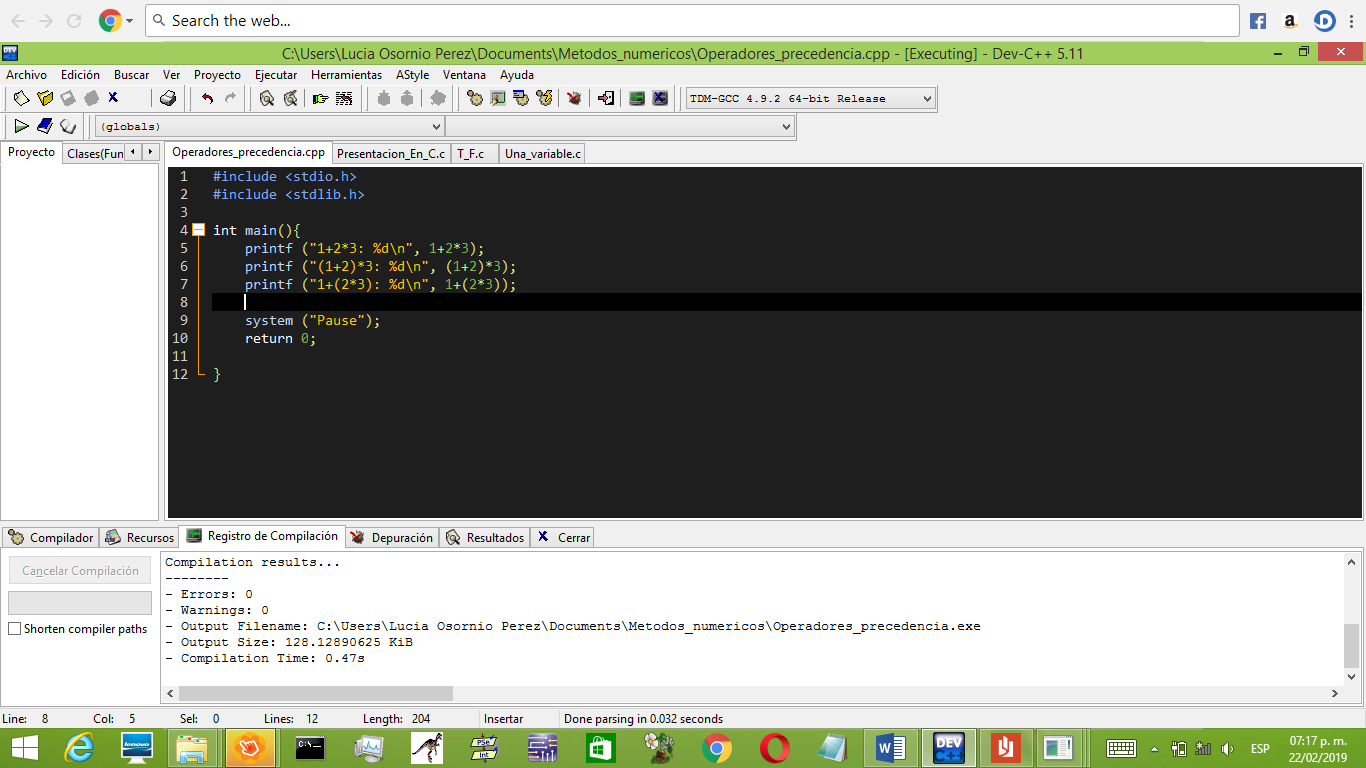
En este programa, se visualizan las operaciones cuyo orden son las operaciones:

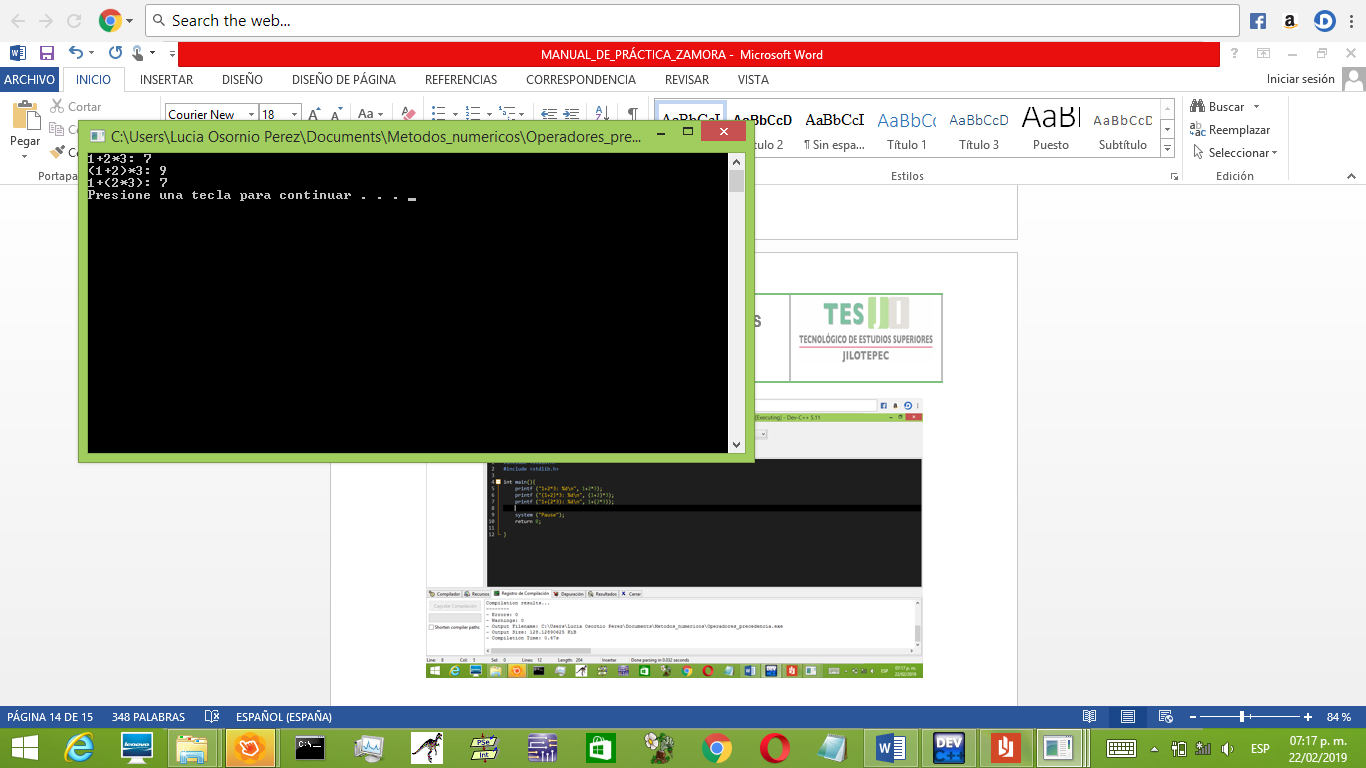
\

\*

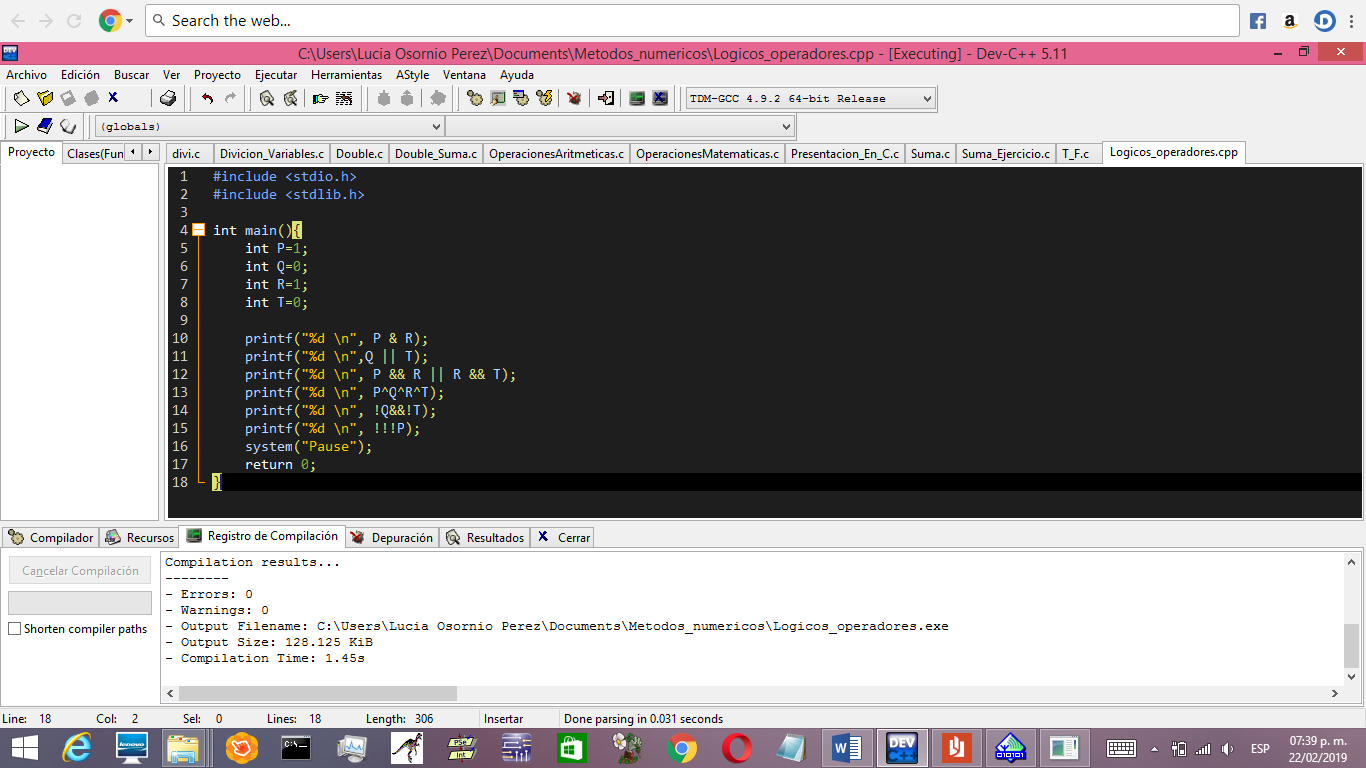
+

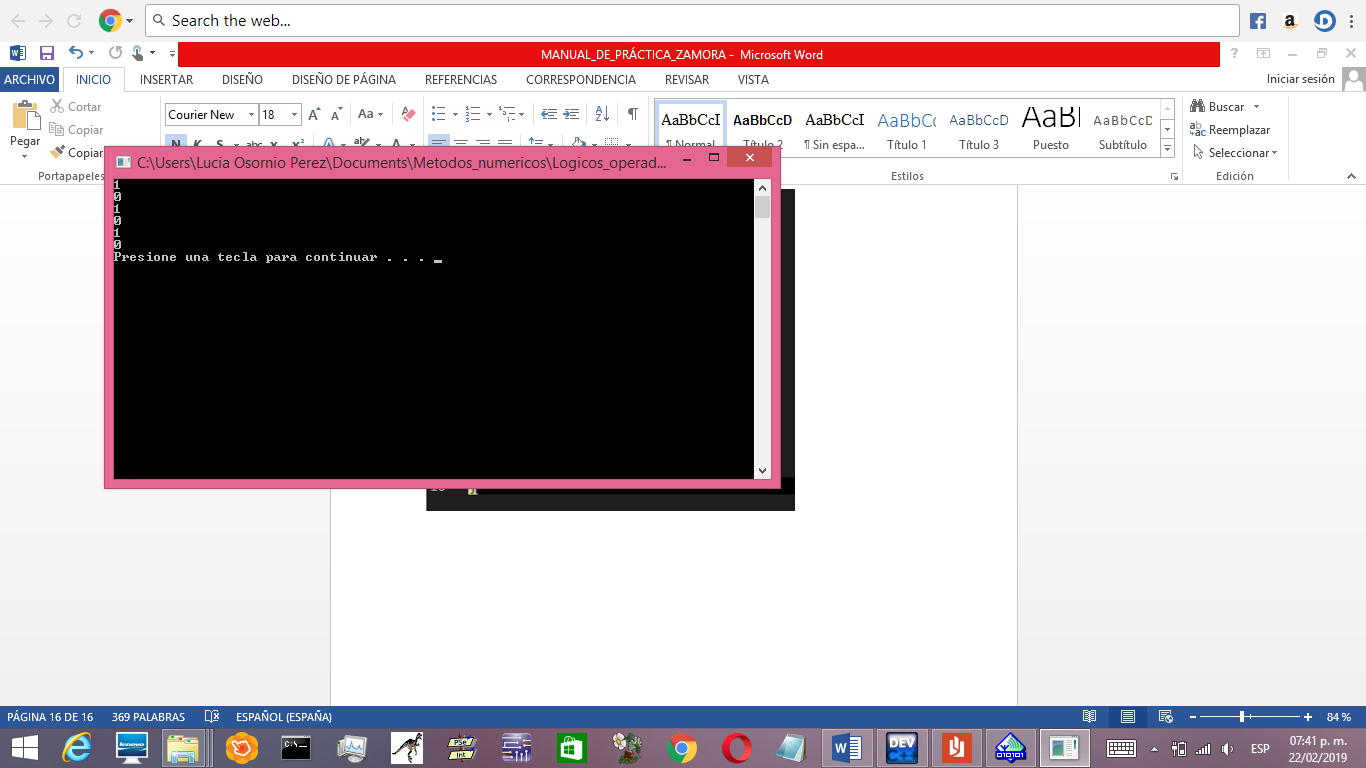
-



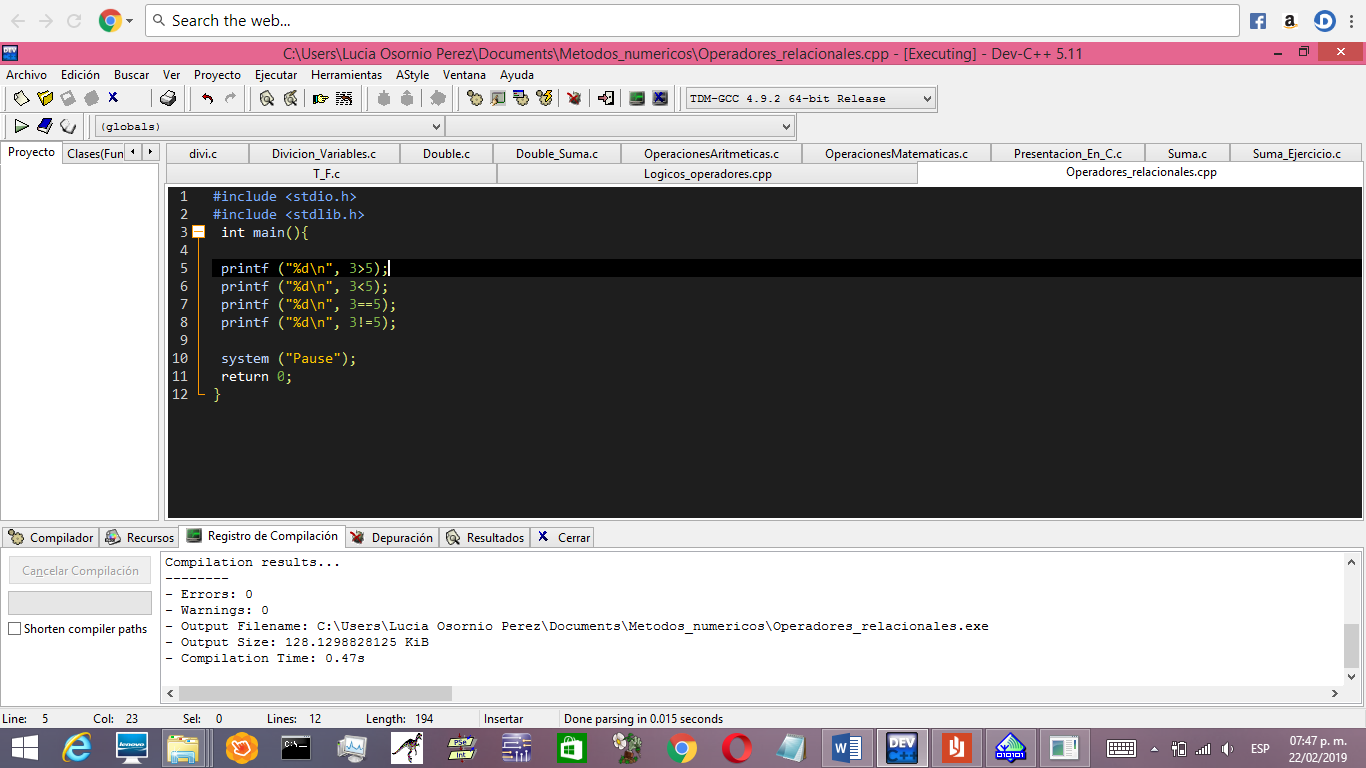


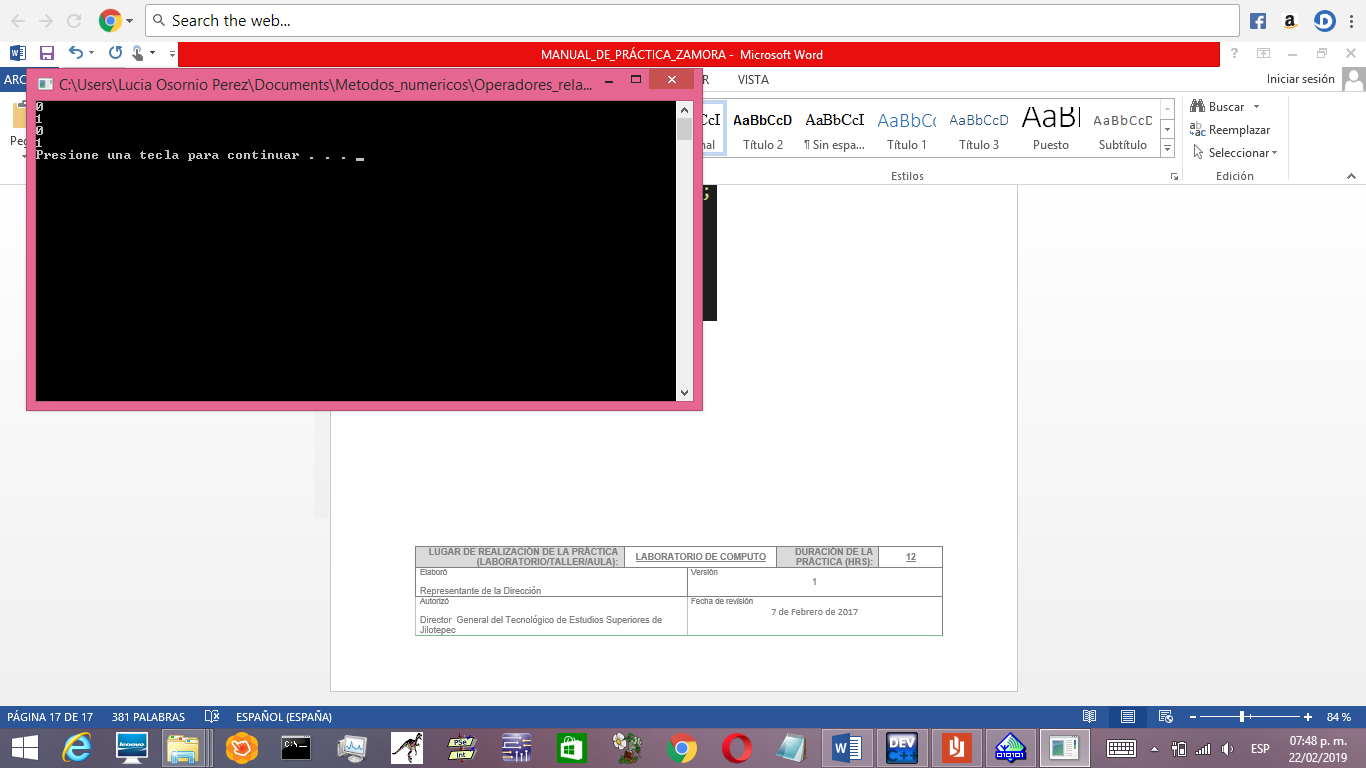
**Operadores lógicos y de relación**

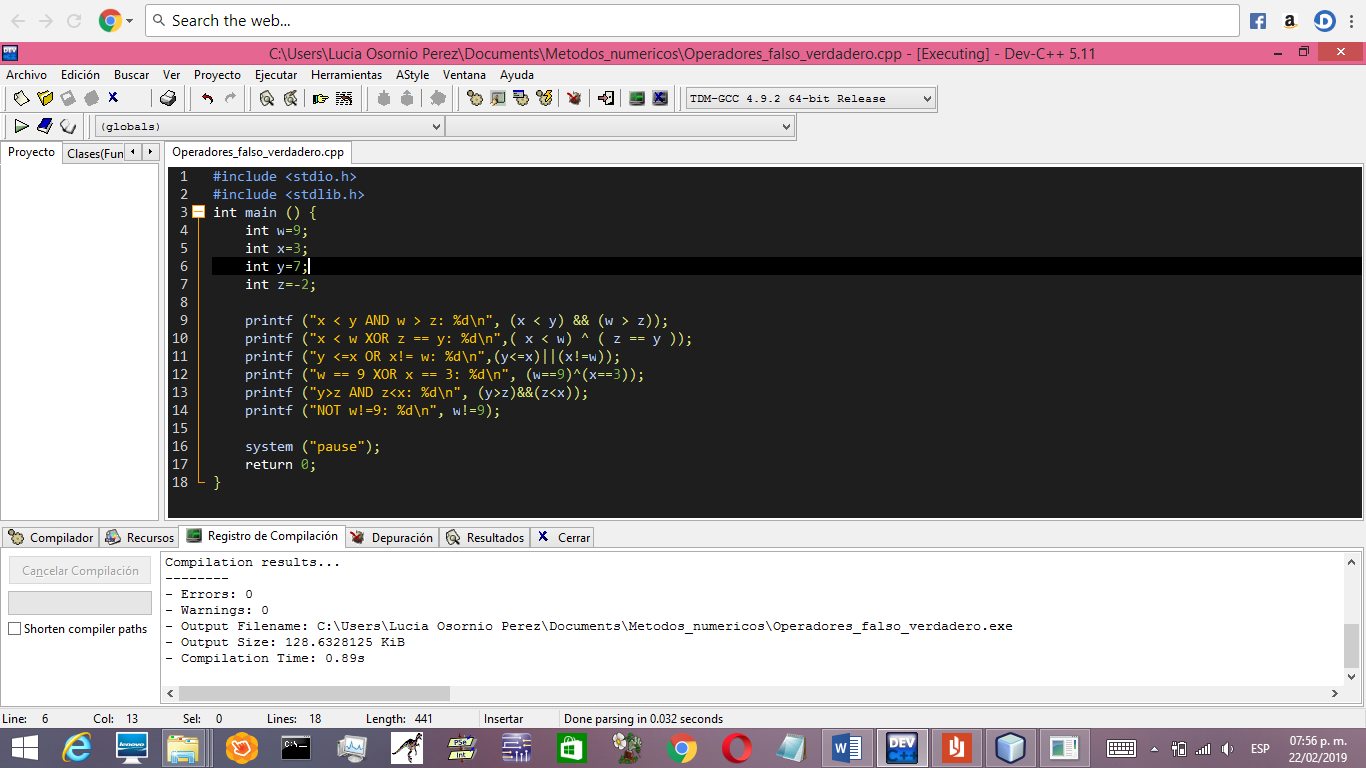


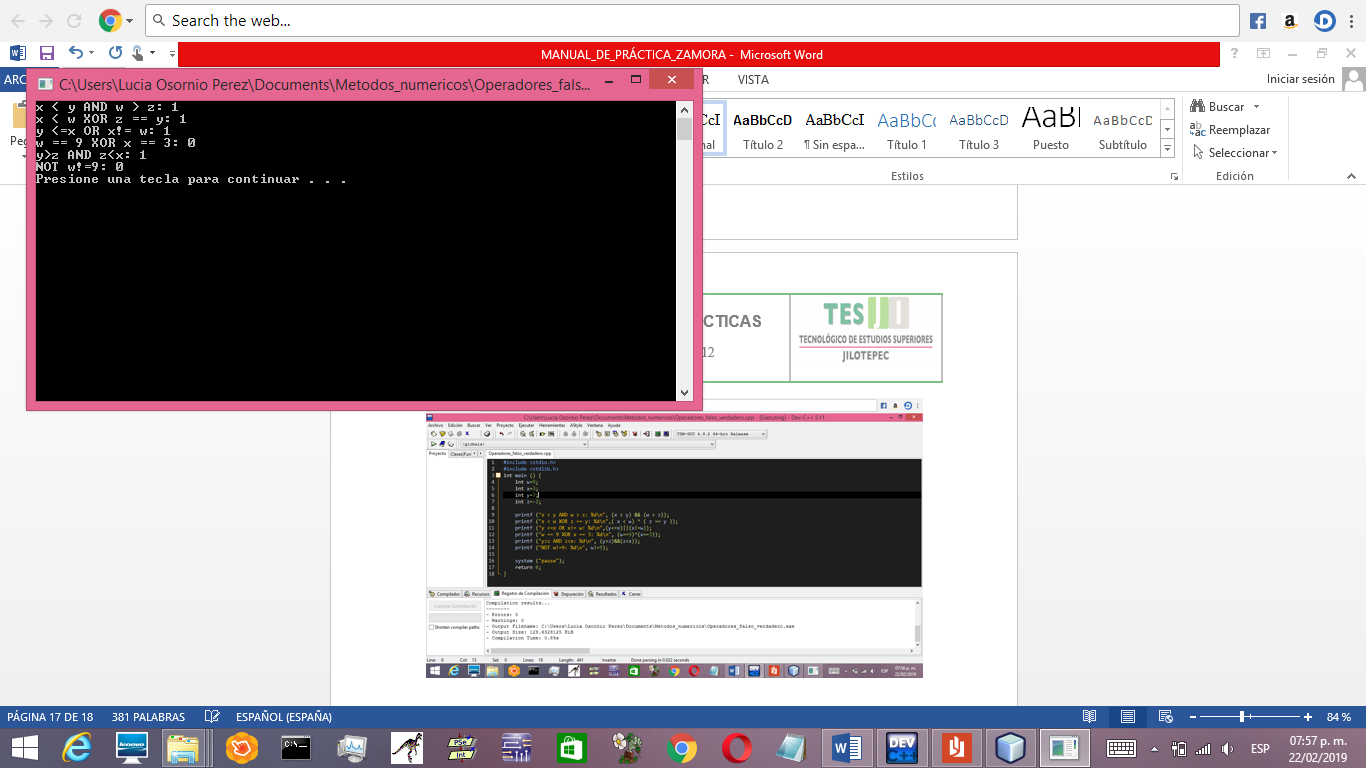


En este programa se visualizan los operadores relacionales. Un ejemplo se muestra









|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **NOMBRE DE LA PRÁCTICA** | **PROGRAMANDO EN LENGUAJE C++ PARTES 2** | | | **No.** | **2** |
| **ASIGNATURA:** | **METODOS NUMERICOS** | **CARRERA:** | **INGENIERÍA EN SISTEMAS COMPUTACIONALES** | **PLAN:** | **ISIC-2010-224** |

**NOMBRE: Lucia Osornio Perez MATRICULA: 201723198 GRUPO: 342**

1. **COMPETENCIA(S) ESPECÍFICA(S):**
2. **MATERIAL EMPLEADO:**

* **C++**
* **LAPTOP**

1. **DESARROLLO DE LA PRÁCTICA:**

**ENTRADA Y SALIDA DE DATOS**

**Imprimir un mensaje simple en pantalla**

**#include <stdio.h>**

**#include <stdlib.h>**

**int main(){**

**printf ("Esto es una impresión en pantalla \n");**

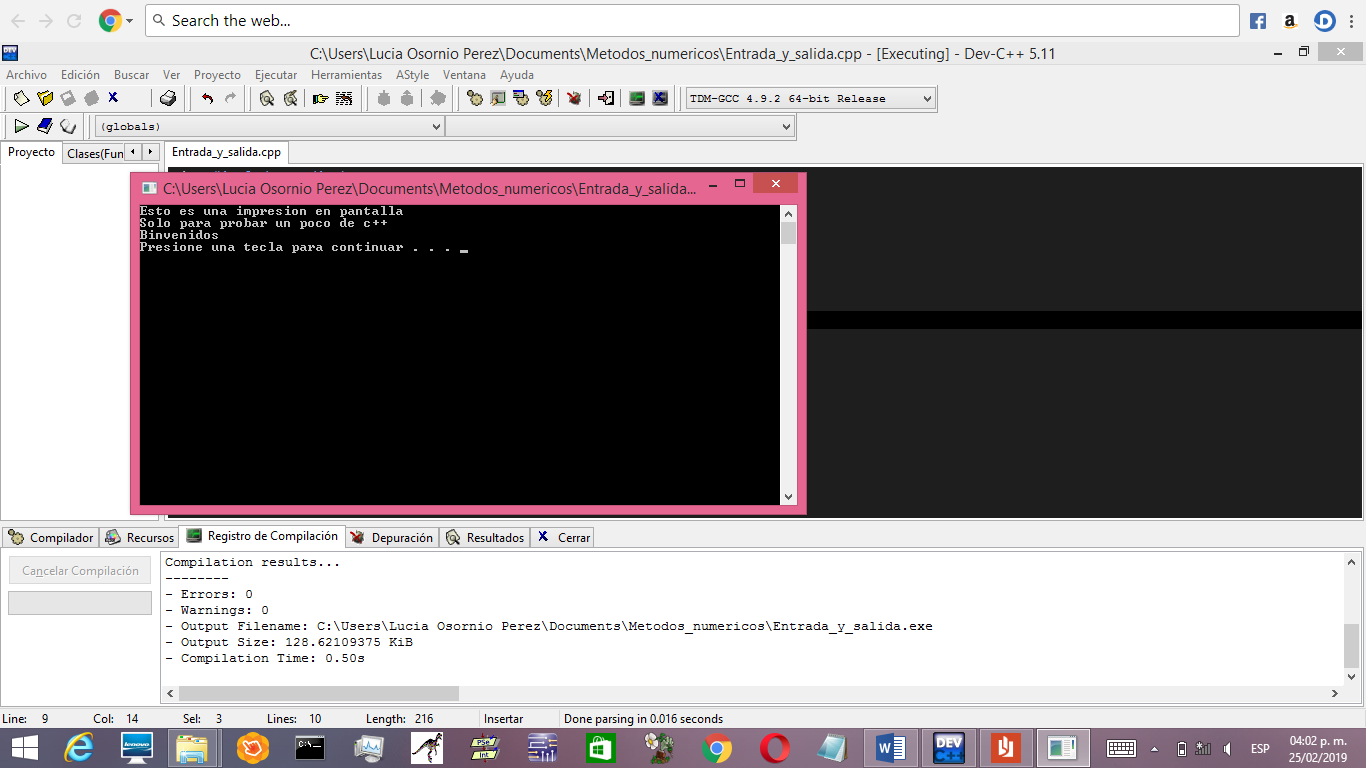
**printf("Solo para probar un poco de c++ \n");**

**printf ("Bienvenidos\n");**

**system("Pause");**

**return 0;**

**}**



**En este programa está en práctica los especificadores de conversión como son %c Un solo carácter**

**%d Entero decimal con signo**

**%f Número decimal de punto flotante**

**%s Cadena de caracteres**

**%u Entero decimal sin signo**

**En la función printf() es la función que acepta una serie de argumentos, aplicándose cada uno de ellos a un especificador de conversión el formato dado. Por cada argumento debe haber un especificador de conversión.**

**Se debe incluir el archivo de encabezado STDIO.H**

**Puts se usa para imprimir mensajes en pantalla, pero no puede desplegar variables numéricas.**

**Toma una sola cadena como su argumento, y la despliega, añadiendo automáticamente una nueva línea al final.**

**#include <stdio.h>**

**#include <stdlib.h>**

**int main(){**

**puts ("El valor de la vida, es la intensidad en que la vives ");**

**system ("Pause");**

**return 0;**

**}**

**Lee datos del teclado de acuerdo a un formato especificado, y asigna los datos de entrada a una o más variables del programa.**

**Usa los mismos especificadores de conversión que la función printf.**

**Puede aceptar más de un valor si se incluyen varios especificadores de conversión y varias variables.**

**IF**

**En este programa se compilan los datos que son pares y los que no los saca del programa**

**Entender e implementar las estructuras de decisión y sus variantes.**

**#include <stdio.h>**

**#include <stdlib.h>**

**int main (){**

**int numero=34;**

**if (numero %2 == 0){**

**printf ("El numero es par \n");**

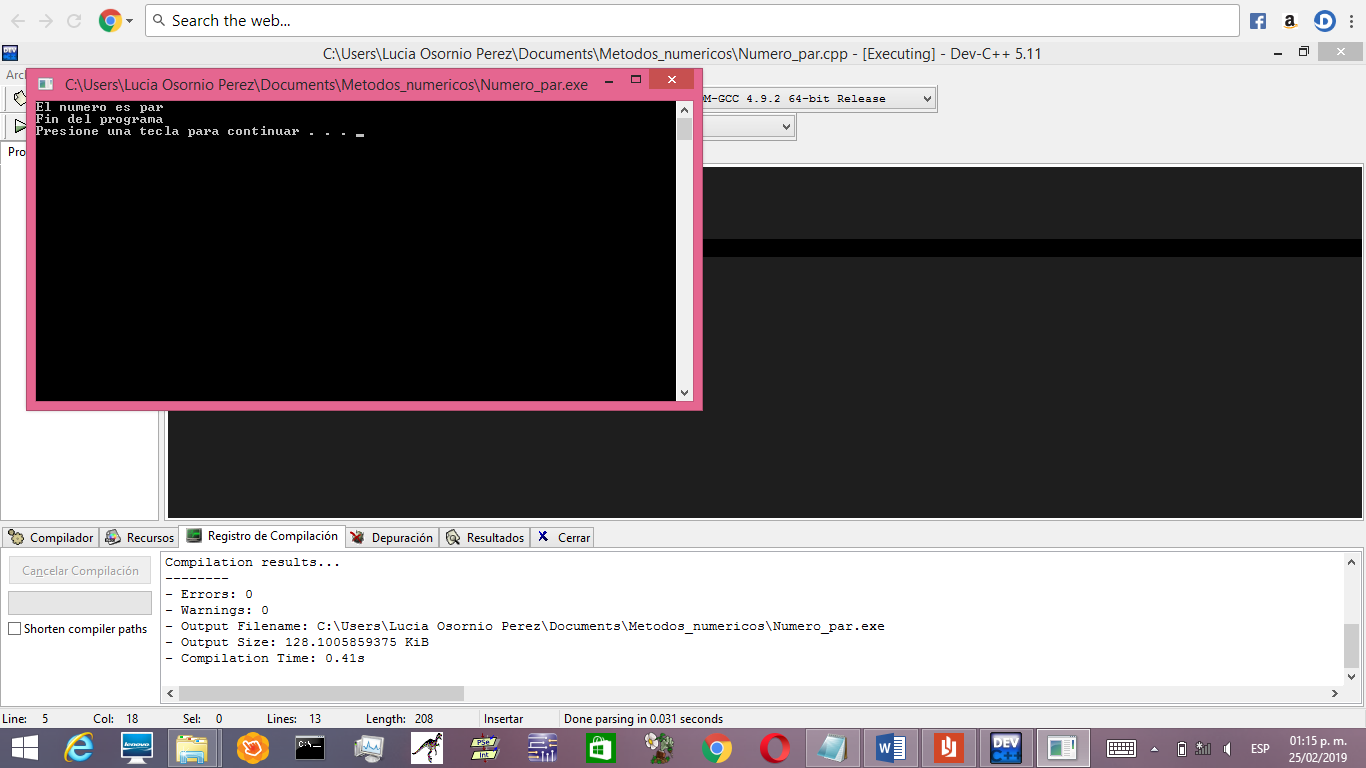
**}**

**printf ("Fin del programa \n");**

**system ("Pause");**

**return 0;**

**}**



**Cuando no es par… el programa corre**

**#include <stdio.h>**

**#include <stdlib.h>**

**int main (){**

**int numero=55;**

**if (numero %2 == 0){**

**printf ("El numero es par \n");**

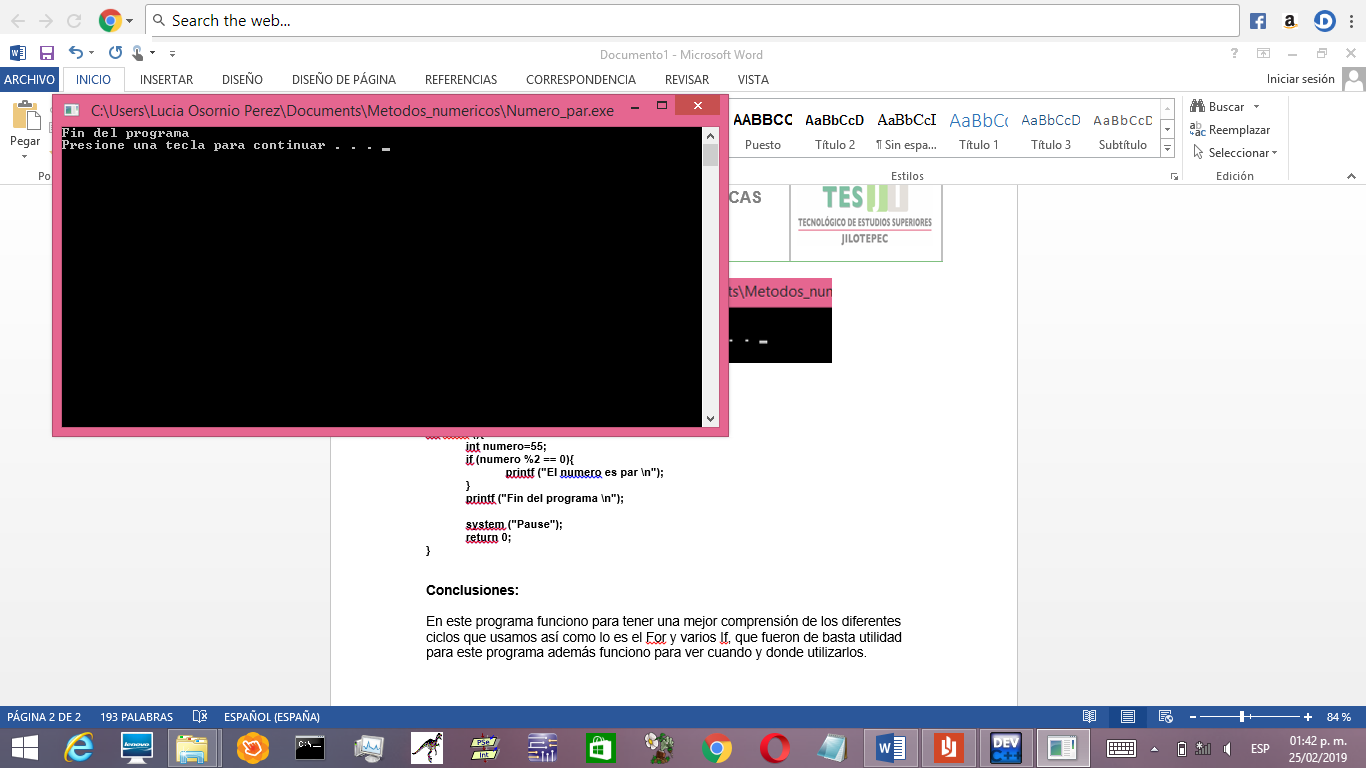
**}**

**printf ("Fin del programa \n");**

**system ("Pause");**

**return 0;**

**}**



**En este programa se visualizan las condiciones cuando se cumplen se imprime sino fin del programa.**

**#include <stdio.h>**

**#include <stdlib.h>**

**int main (){**

**int numero=55;**

**if (numero < 100){**

**printf ("El numero es menor que 100 \n");**

**if (numero > 50)**

**printf ("y mayor que 50 \n");**

**}**

**printf ("Fin del programa \n");**

**system ("Pause");**

**return 0;**

**}**



**Este es el caso cuando la condición es negativa.**

**#include <stdio.h>**

**#include <stdlib.h>**

**int main (){**

**int numero=35;**

**if (numero < 100){**

**printf ("El numero es menor que 100 \n");**

**if (numero > 50)**

**printf ("y mayor que 50 \n");**

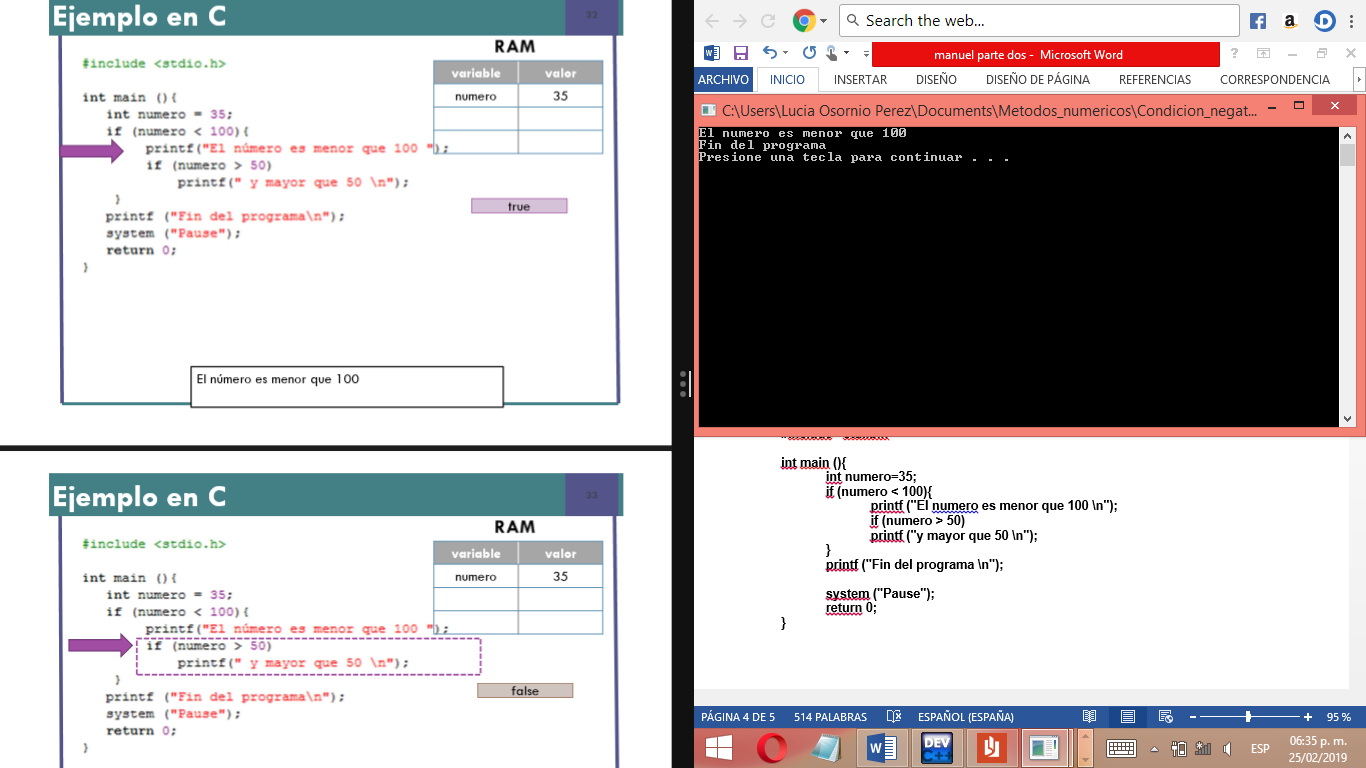
**}**

**printf ("Fin del programa \n");**

**system ("Pause");**

**return 0;**

**}**



**En este caso la variable no cumple con la primera condición por la tanto no se imprime**

**#include <stdio.h>**

**#include <stdlib.h>**

**int main (){**

**int numero=135;**

**if (numero < 100){**

**printf ("El numero es menor que 100 \n");**

**if (numero > 50)**

**printf ("y mayor que 50 \n");**

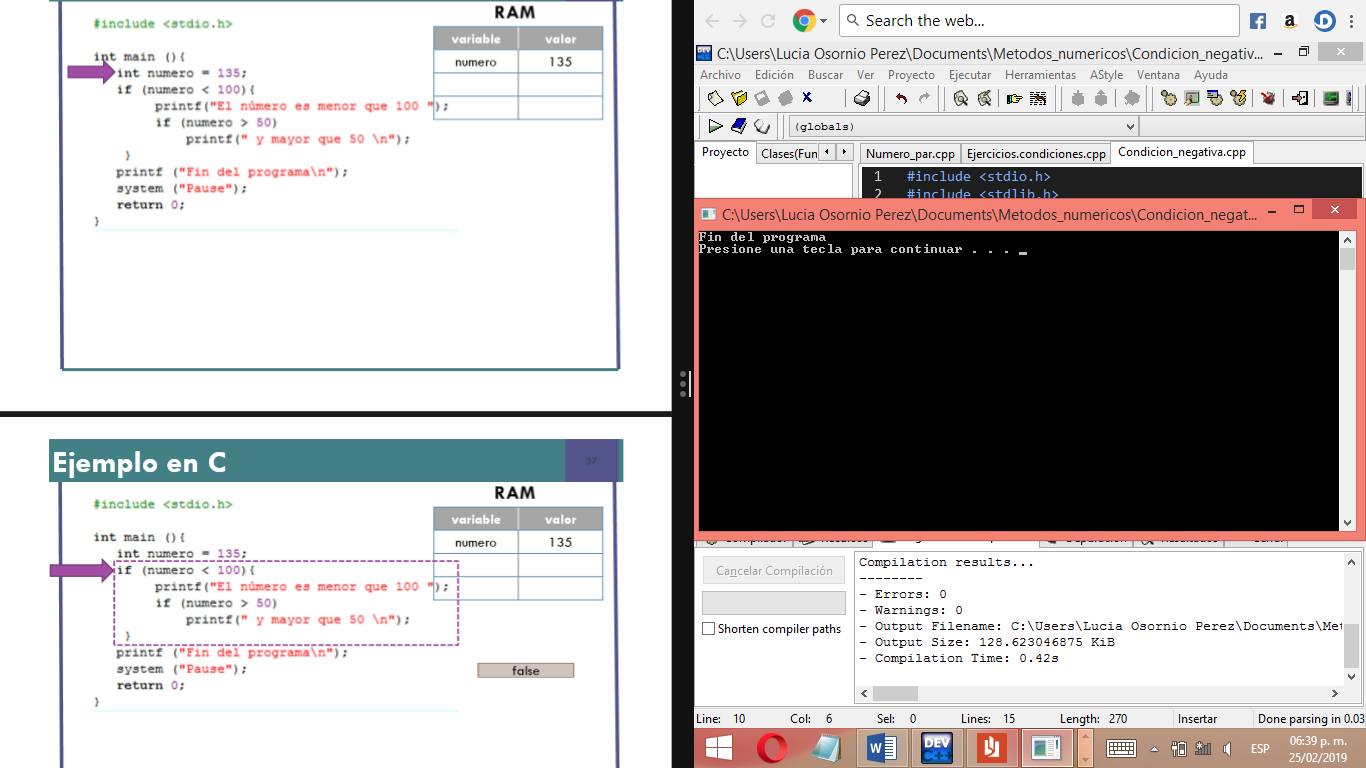
**}**

**printf ("Fin del programa \n");**

**system ("Pause");**

**return 0;**

**}**



**Un ejercicio sobre las calificaiones es**

**#include <stdio.h>**

**#include <stdlib.h>**

**int main (){**

**int cali;**

**puts ("Ingresa la calificacion de tu examen");**

**scanf ("%d", &cali);**

**if (cali >= 8){**

**puts ("¡¡Felicidades aprobaste el curso!!");**

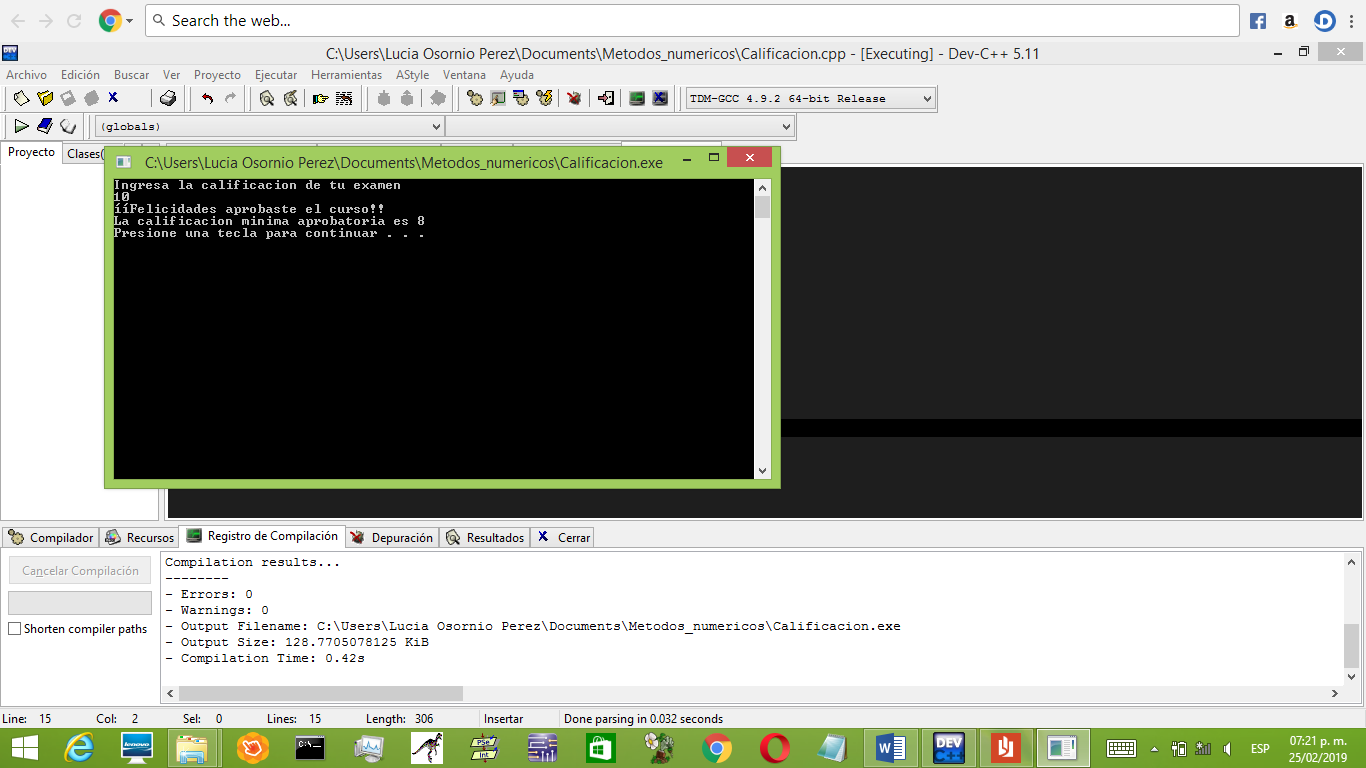
**}**

**printf ("La calificacion minima aprobatoria es 8 \n");**

**system ("Pause");**

**return 0;**

**}**



**En este caso, se visualiza un if con un else de acompañamiento, es decir, un si pero con un pero si.**

**#include <stdio.h>**

**#include <stdlib.h>**

**int main (){**

**int numero= 36;**

**if (numero < 50){**

**printf ("El numero es menor que 50 \n");**

**}else{**

**printf ("El numero es mayor que 50 \n");**

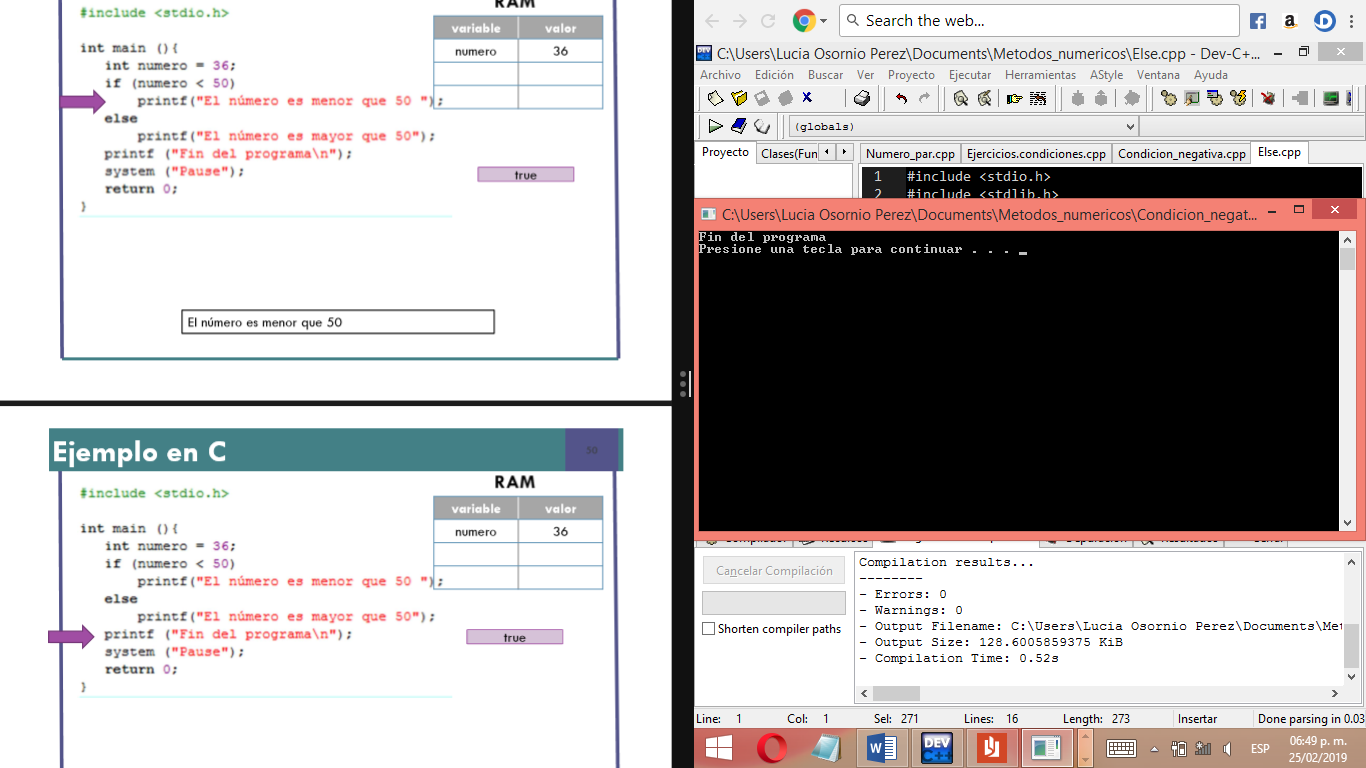
**}**

**printf ("Fin del programa \n");**

**system ("Pause");**

**return 0;**

**}**



**En el caso contrario es**

**#include <stdio.h>**

**#include <stdlib.h>**

**int main (){**

**int numero= 98;**

**if (numero < 50){**

**printf ("El numero es menor que 50 \n");**

**}else{**

**printf ("El numero es mayor que 50 \n");**

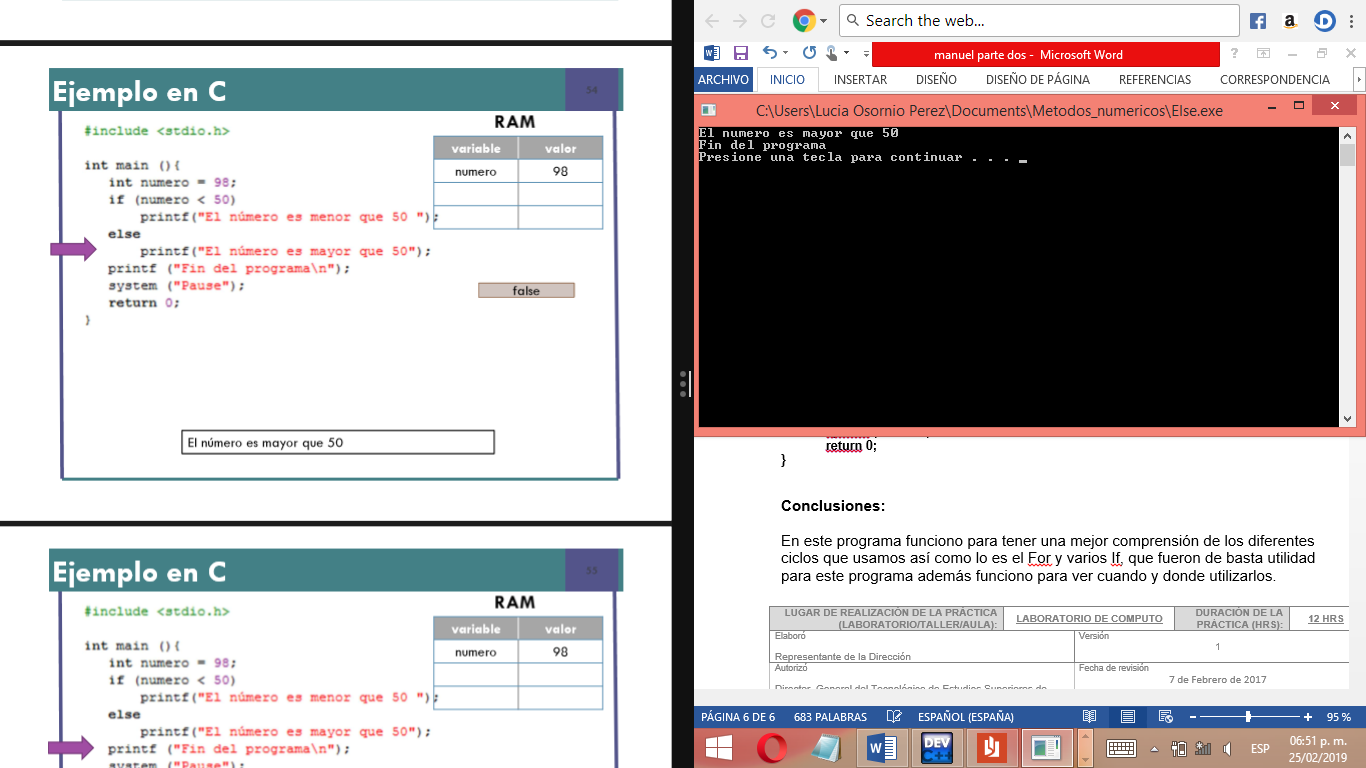
**}**

**printf ("Fin del programa \n");**

**system ("Pause");**

**return 0;**

**}**



**En este programa se ve claro, que las condiciones también pueden ser cumplidas con variables… un ejemplo es el descuento.**

**//se recuerda que 1 es verdadero**

**//Mientras que 0 es negativo**

**#include <stdio.h>**

**#include <stdlib.h>**

**int main (){**

**int descuento =10;**

**int descuentoCliente=20;**

**//Falso =0, verdadero=1**

**int cliente =1;**

**int z=0;**

**//Si eres cliente frecuente, tiene 20% de descuento**

**//Sino eres cliente frecuente 0, solo tiene el 10% de descuento**

**z=(cliente) ? descuentoCliente : descuento;**

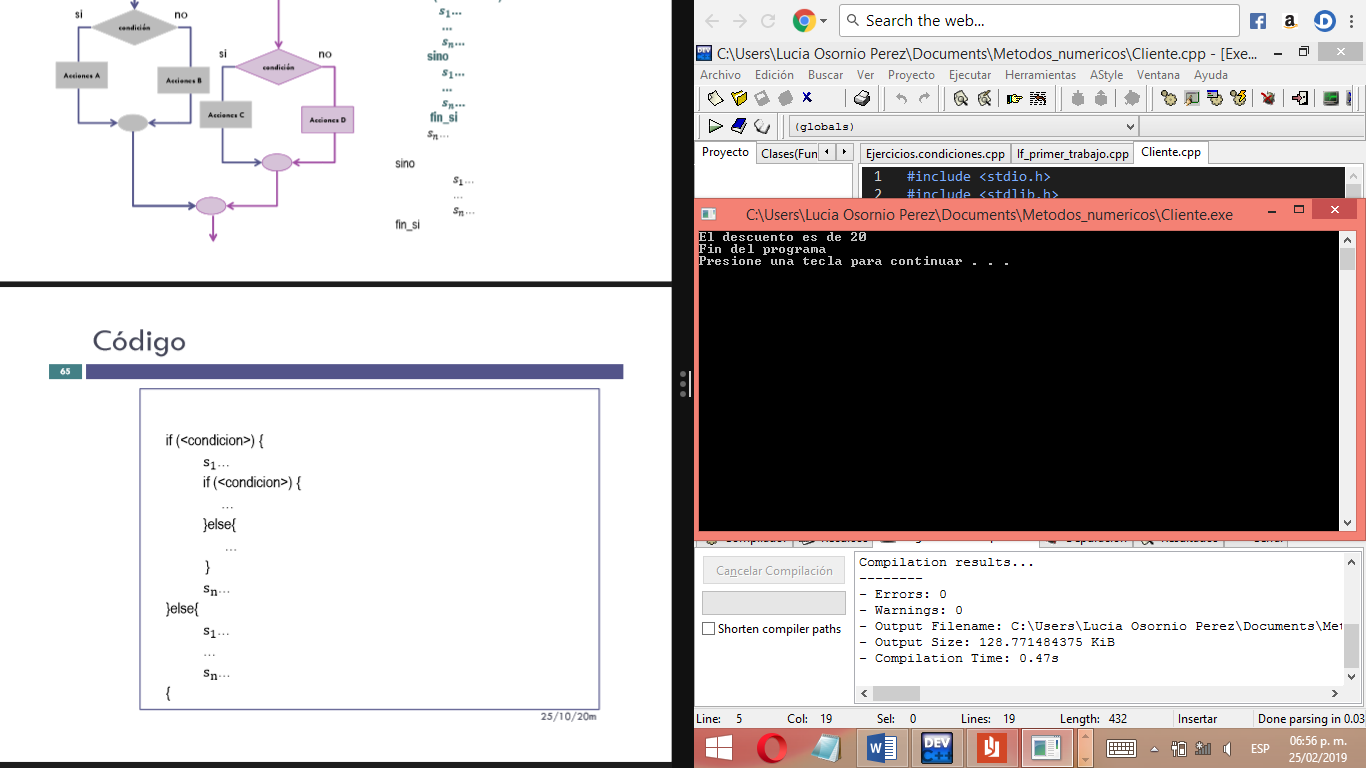
**printf ("El descuento es de %d \% \n", z);**

**printf ("Fin del programa \n");**

**system ("Pause");**

**return 0;**

**}**



**WHILE**

**El propósito de este ciclo es repetir un bloque de código mientras una condición se mantenga verdadera.**

**Verifica si la condición se cumple**

**Si es verdadero, ejecuta una o varias instrucciones y nuevamente verifica la condición.**

**Si es falsa, entonces el ciclo termina.**

**#include <stdio.h>**

**#include <stdlib.h>**

**int main (){**

**int contador=0;**

**while (contador < 3){**

**printf ("Hola \n");**

**contador++;**

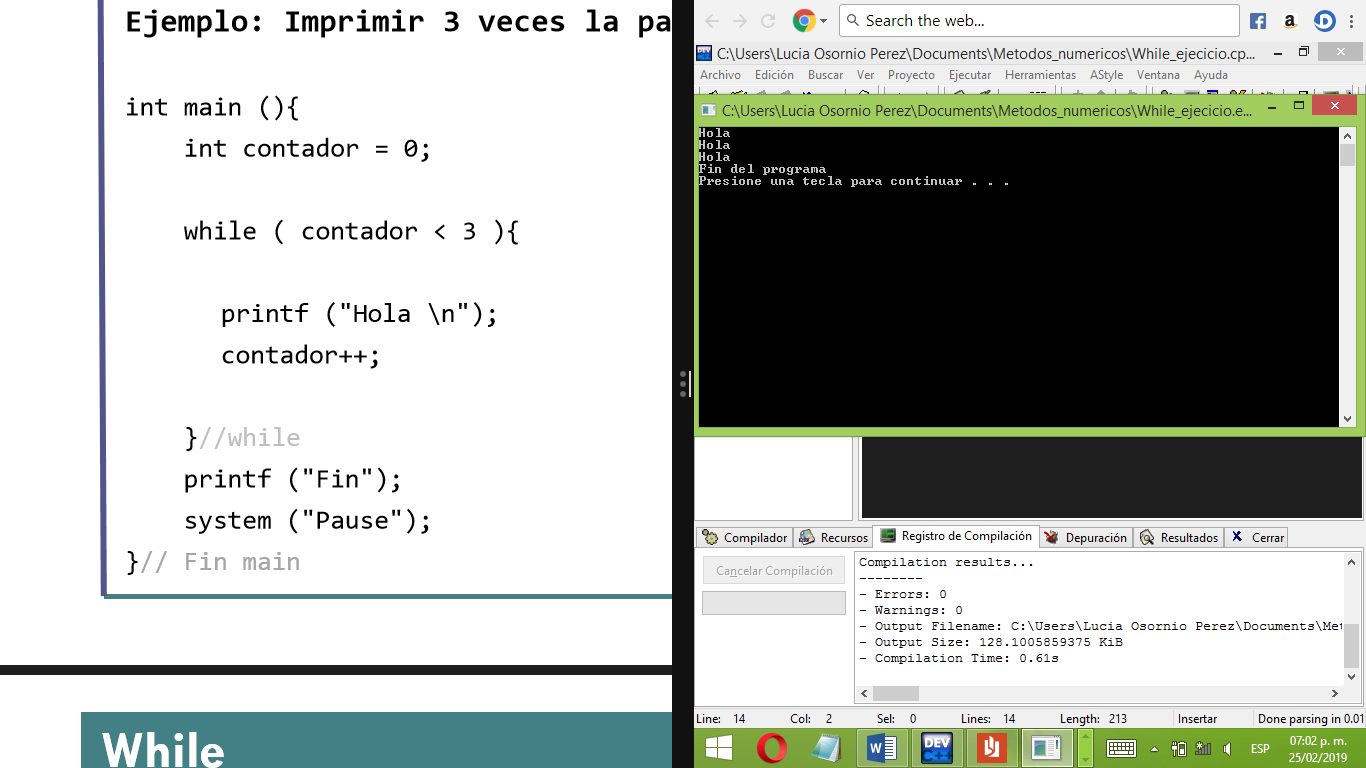
**}**

**printf ("Fin del programa \n");**

**system ("Pause");**

**return 0;**

**}**



**En el caso contrario se muestra como fin del programa.**

**#include <stdio.h>**

**#include <stdlib.h>**

**int main (){**

**int contador=5;**

**while (contador < 3){**

**printf ("Hola \n");**

**contador++;**

**}**

**printf ("Fin del programa \n");**

**system ("Pause");**

**return 0;**

**}**

**Conclusiones:**

Estos problemas trataron de cómo y cuándo se debe utilizar el if, while y está por verse el for. Sin embargo, el tema principal nos ayudó a conocer y saber el porqué de las variables de entrada y salida.

En los programas se hicieron prácticas de cómo se compilaba si la condición es verdadera…también hubo casos donde se colocaba la condición contraria para ver si se compila.

Como conclusión general podemos, decir, que el while es para casos repetitivos cumpliéndose con las condiciones, mientras que el if es para condiciones tras condiciones.