|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre de la práctica** | **LAMINAS DE LENGUAJE C** | | | **No.** | **1** |
| **Asignatura:** | **Métodos numéricos** | **Carrera:** | **Ing. Sistemas Computacionales** | **Duración de la práctica (Hrs)** | **2** |

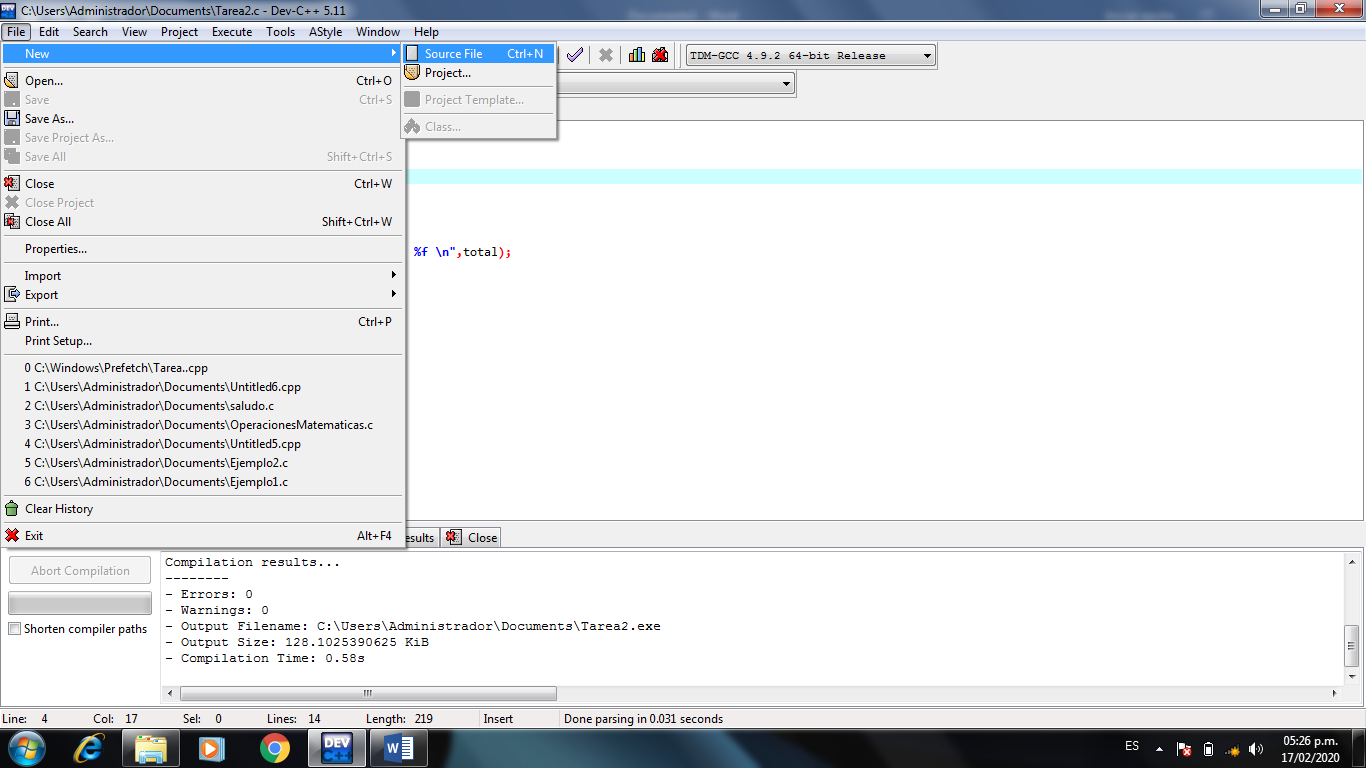
1. **Lugar de realización de la práctica (laboratorio, taller, aula u otro):** Otro.

**III. Material empleado:** laminas y laptop

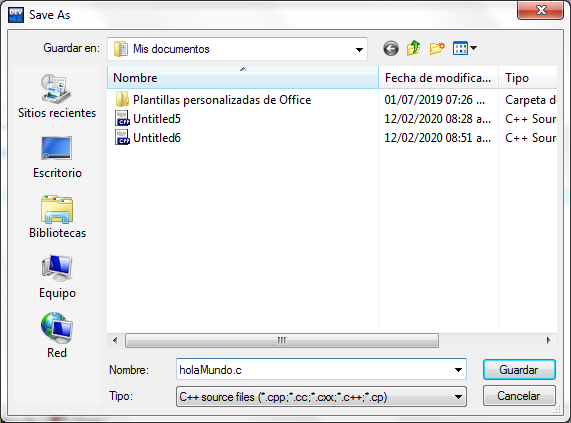
1. **Desarrollo de la práctica:**

# Lamina2 lenguaje C

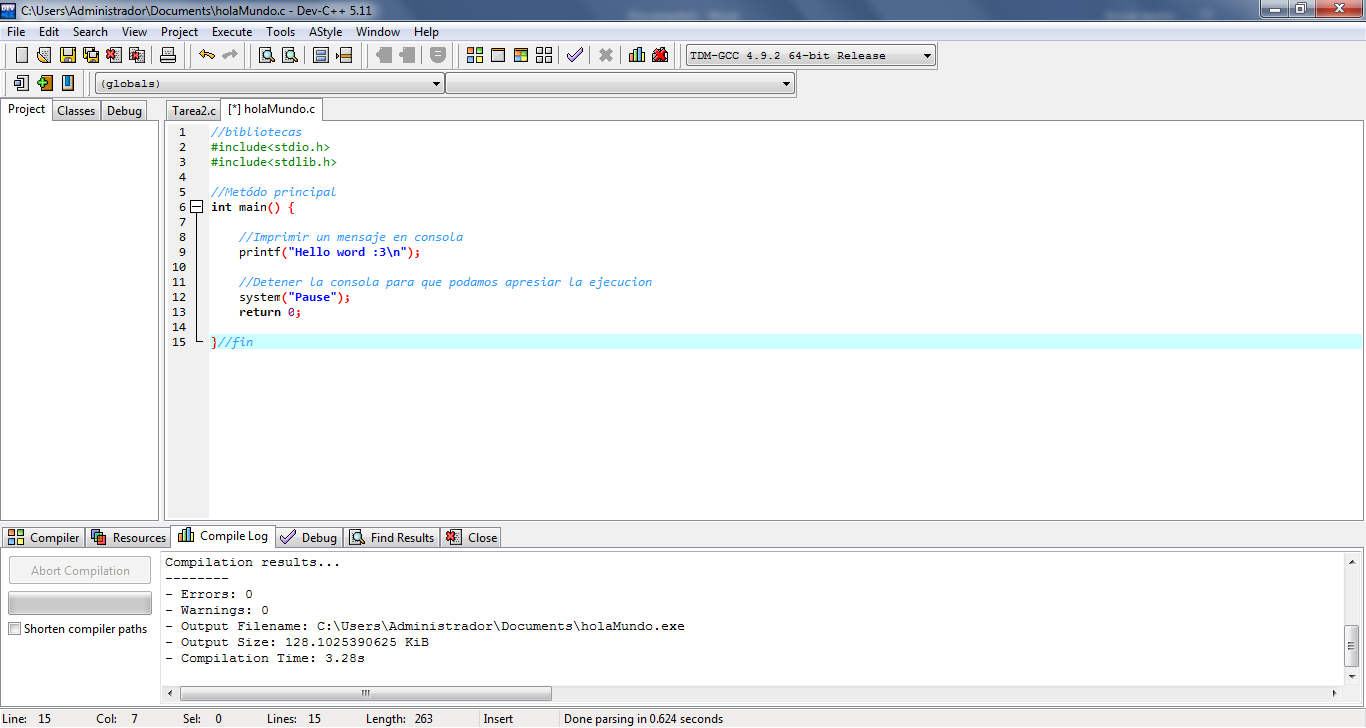
En este primer ejercicio abrimos nuestro programa y creamos un nuevo proyecto dando clic sobre file que se ubica en la parte superior de la barra de herramientas, posteriormente new y source file, como se muestra en la siguiente captura.



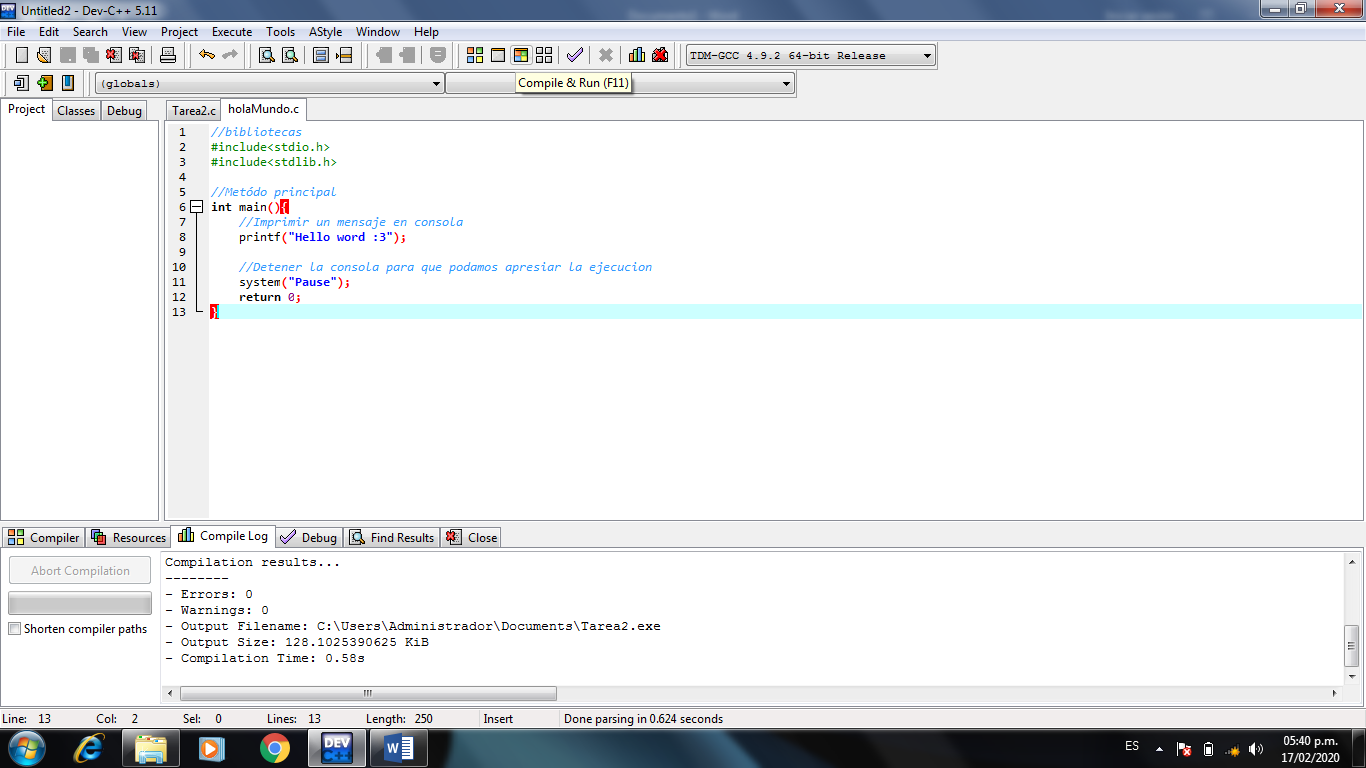
A continuación, Guardamos nuestro proyecto, dando clic nuevamente en File, pero en esta ocasión damos clic sobre Save As y seleccionamos la ruta donde se desea almacenar, además de ello colocamos que el archivo se guarde como .c



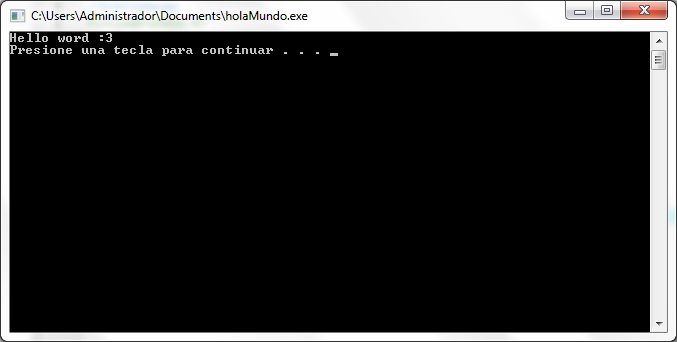
Posteriormente el área de trabajo iniciamos colocando las bibliotecas, las cuales proveen lo elementos necesarios para el desarrollo, compilación y ejecución de programa, después desarrollamos el programa, como se muestra a continuación.



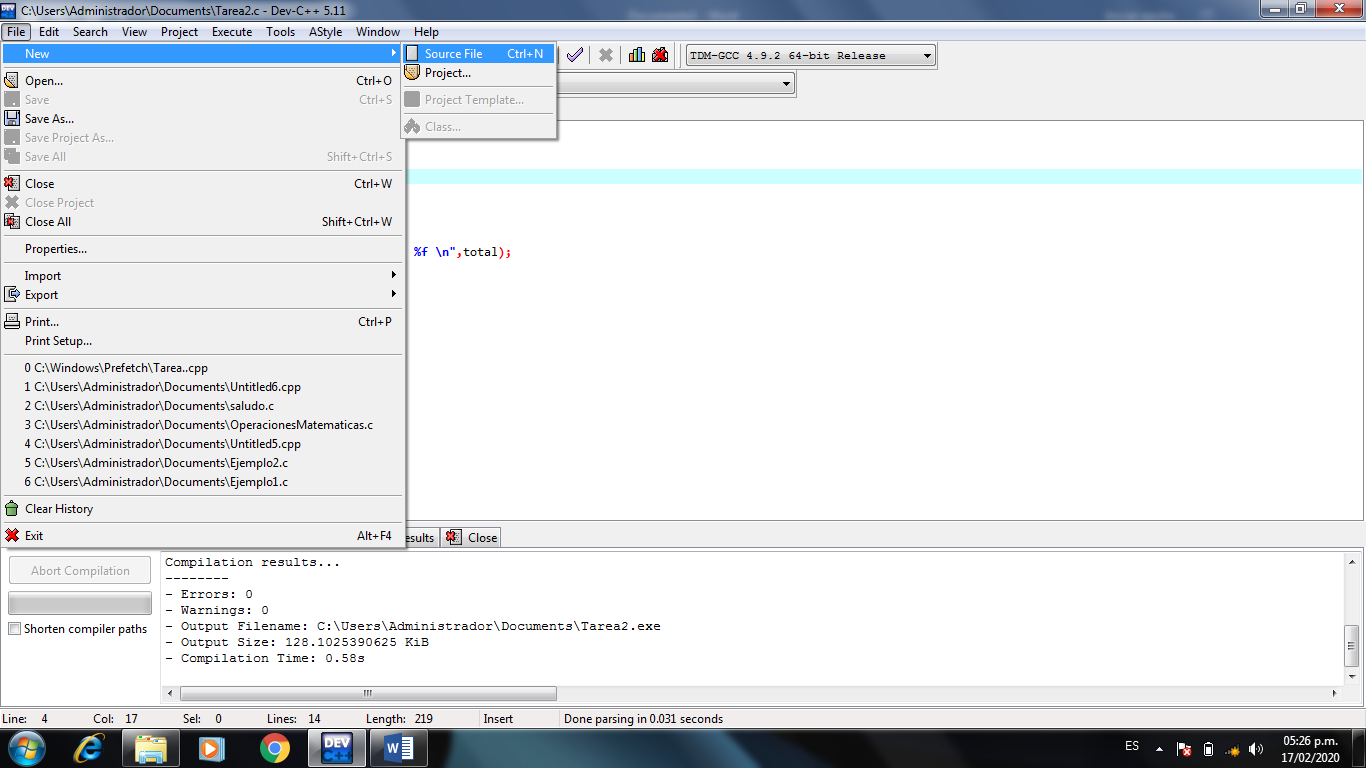
Damos clic sobre el icono para ejecutar el programa

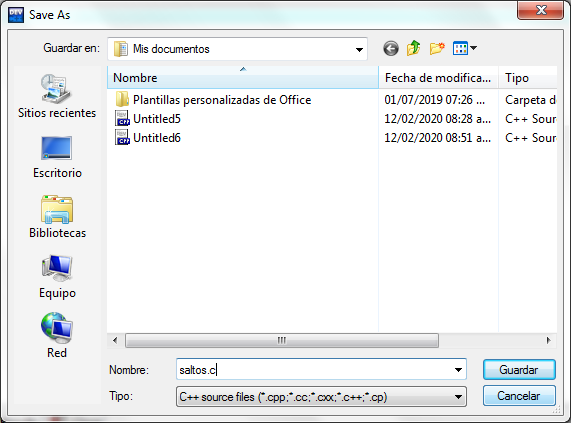


Finalmente, así es como se muestra la ejecución en consola

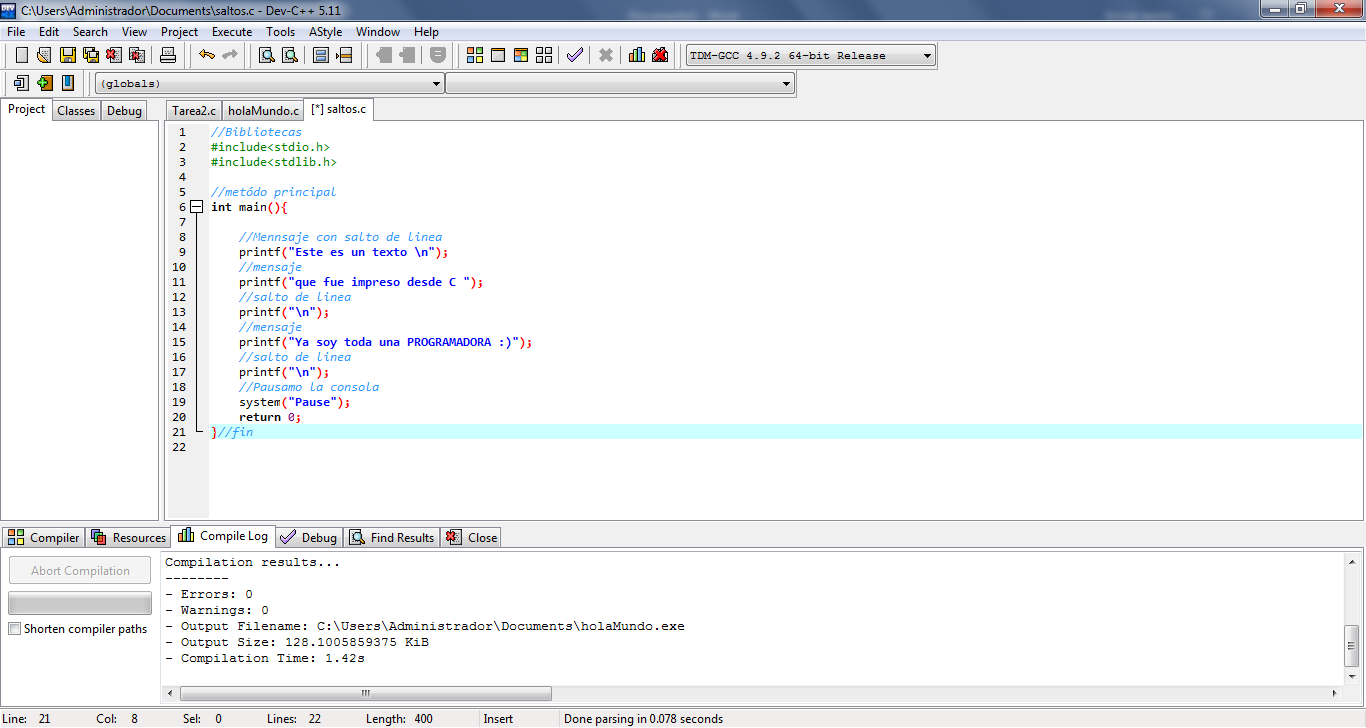


Creamos un nuevo proyecto como anteriormente lo hicimos, con la finalidad de mostrar cómo se hacen los saltos de línea en consola.

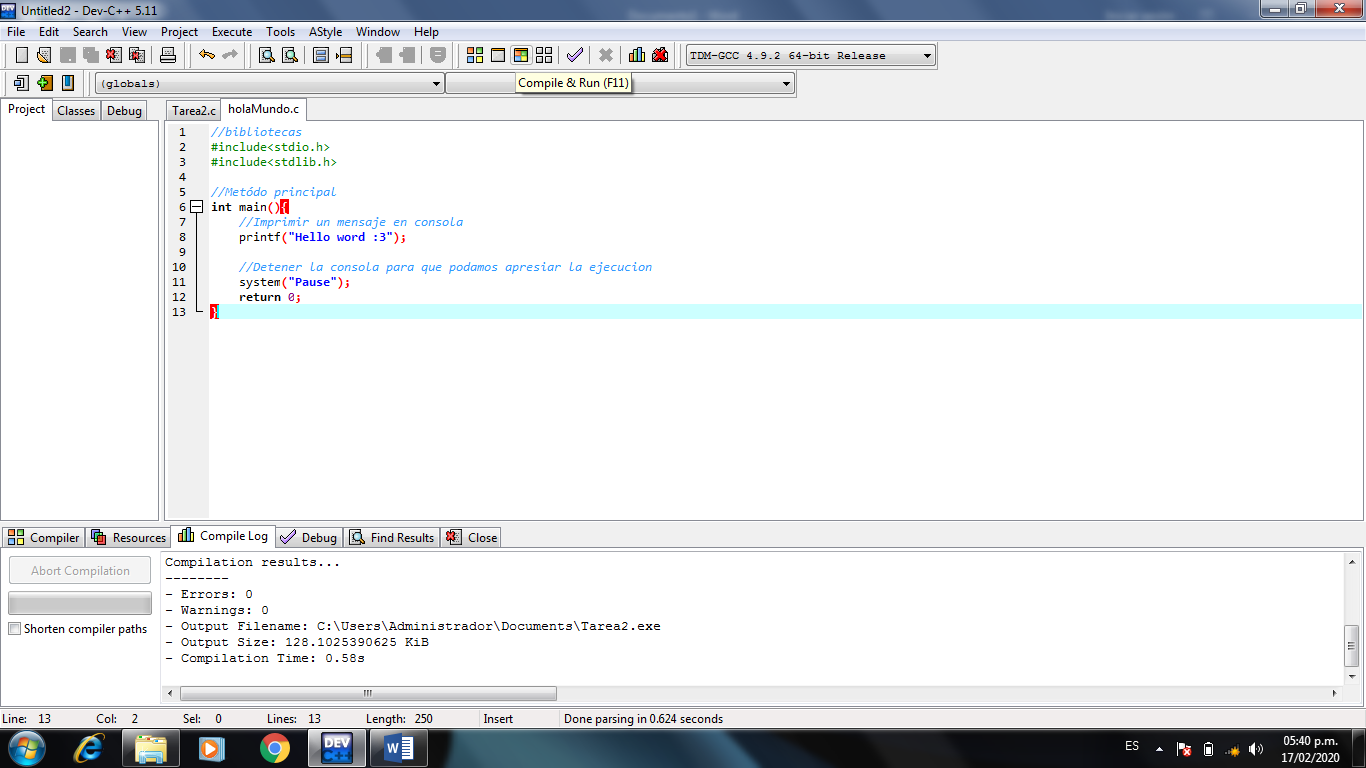


Para comenzar guardamos nuestro proyecto con el nombre de Saltos.c como se muestra 

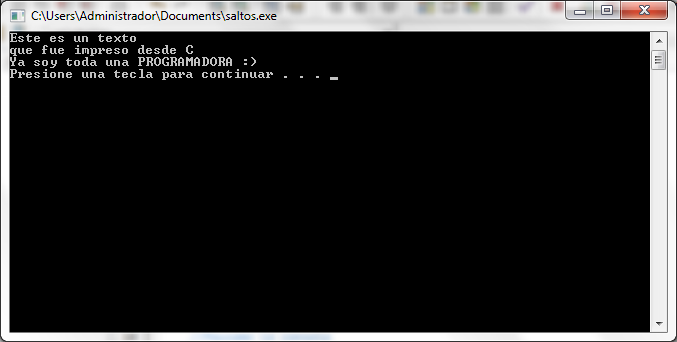
Una vez dentro del área de trabajo se programaron las siguientes líneas de código dentro del método principal, y como al principio lo hicimos se declaran primero las bibliotecas



Presionamos el icono para ejecutar



Así se muestra la ejecución

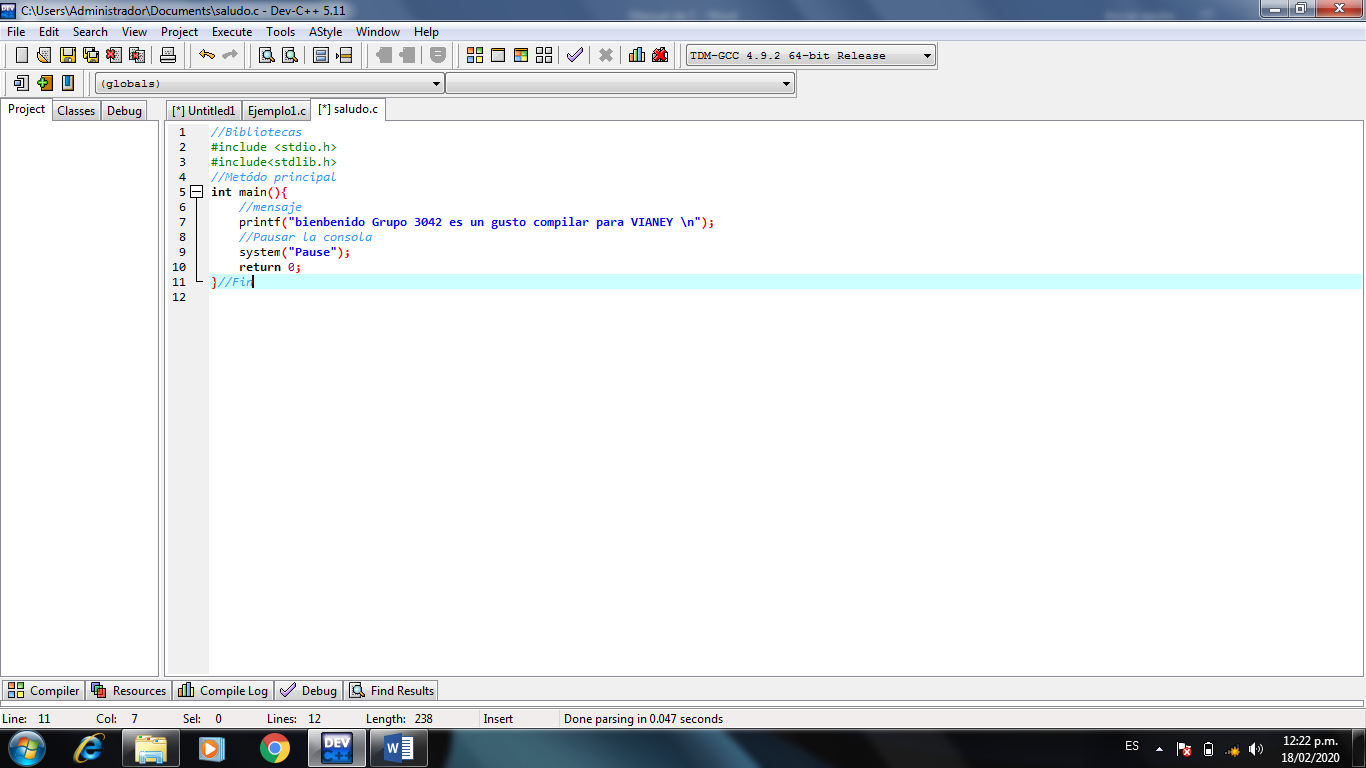


EJERCICIO

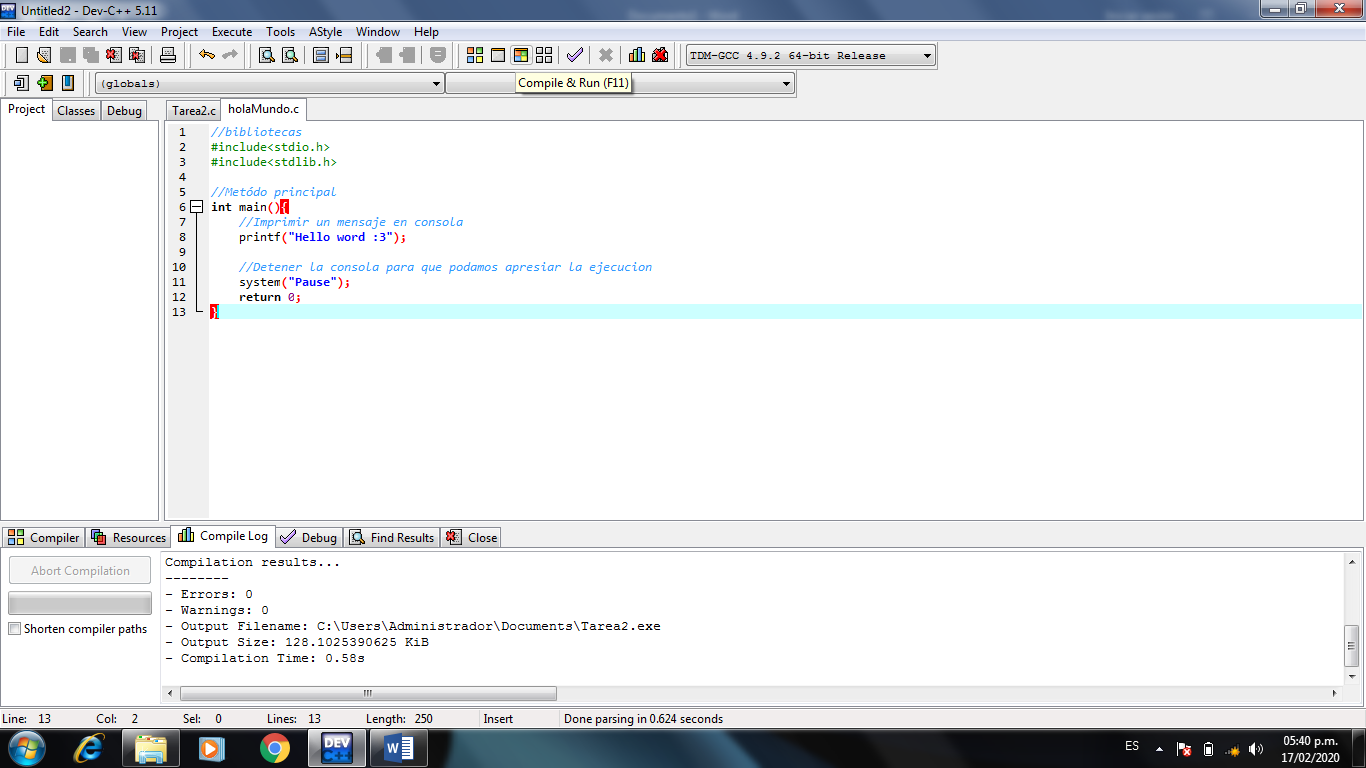
Crea un archivo en Dev – C++

Crea un programa donde imprima la bienvenida con tu nombre.

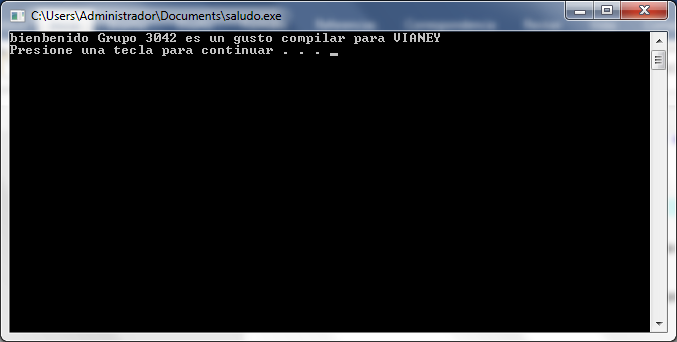
Para el desarrollo de nuestro ejercicio, abrimos un nuevo proyecto el cual guardamos como saludo.c como anteriormente se ha realizado, este es el código que se desarrolló dentro del espacio de trabajo.



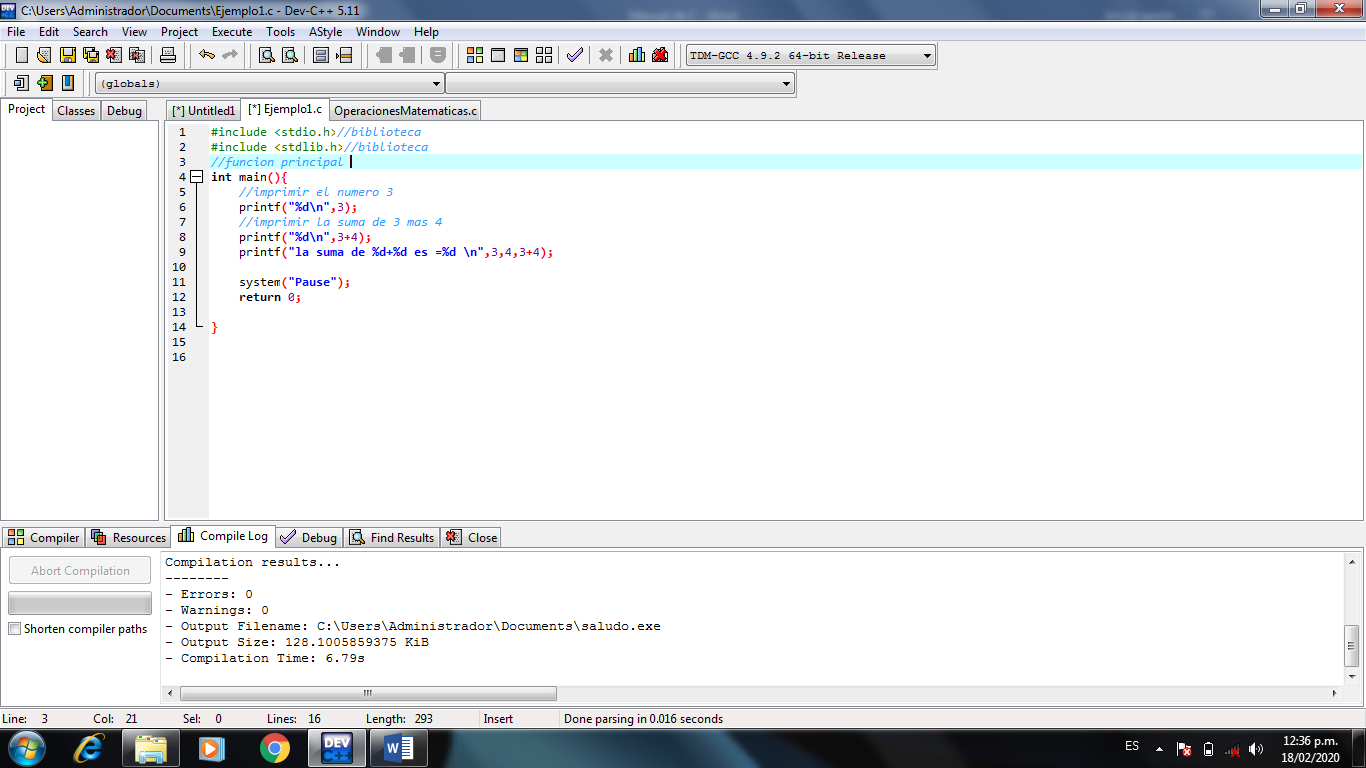
Para ejecutar el programa, nos colocamos en la barra de herramientas en la parte superior de la pantalla y presionamos el siguiente icono



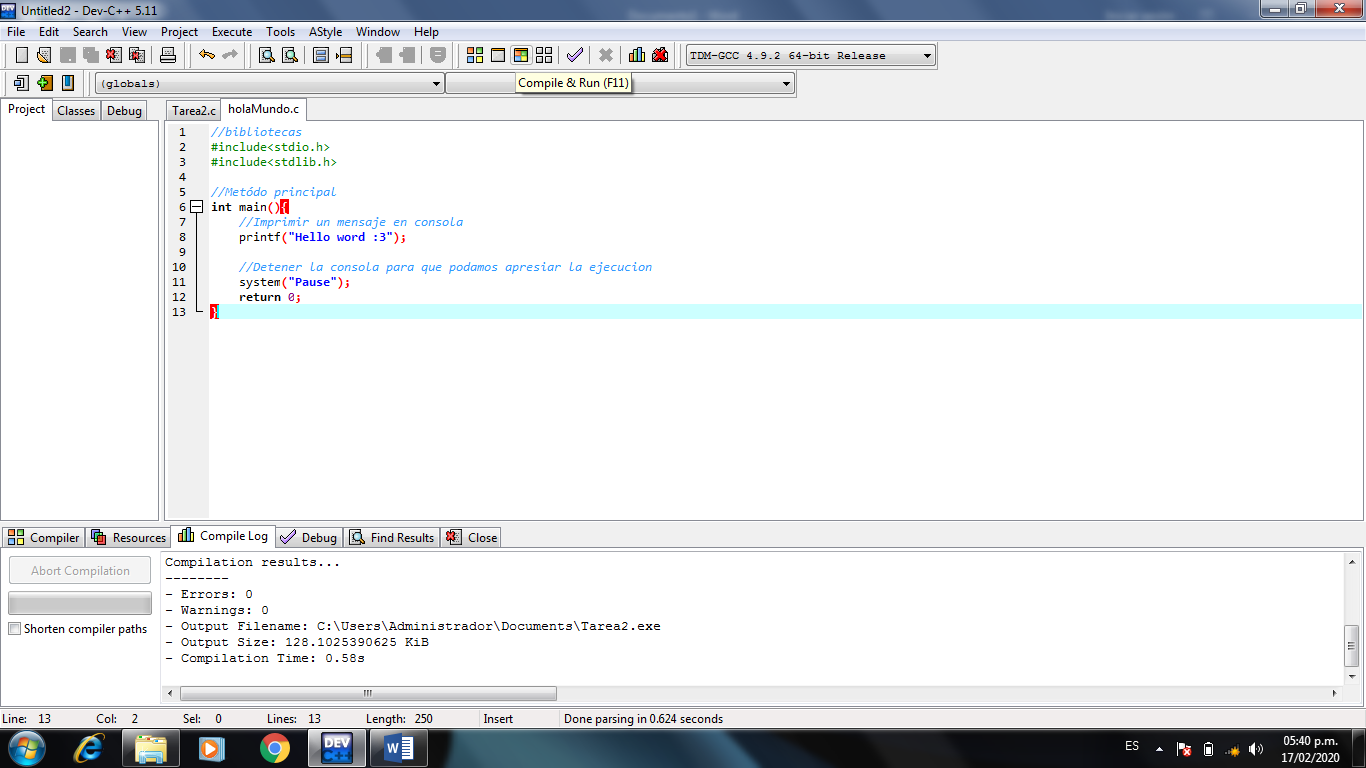
Y a continuación se muestra nuestro mensaje anteriormente programado



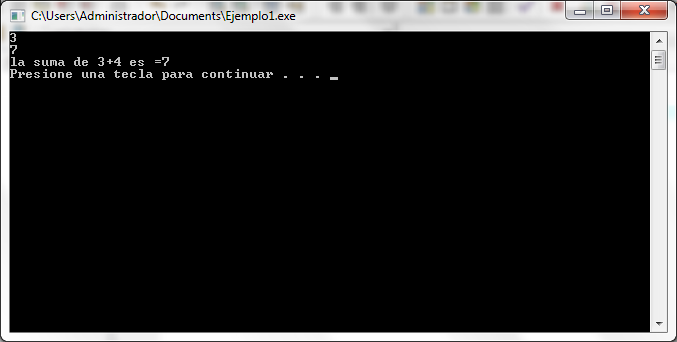
El siguiente programa, será ahora para mostrar el uso de los datos numéricos, como se muestra a continuación, lo que realizamos es crear un nuevo programa y guardarlo con un nombre que haga alusión a la fusión que realiza, en este caso será Ejemplo1, ya que es el primer ejemplo que se realiza de operaciones, el código desarrollado es el que se muestra a continuación



Ejecutamos como anteriormente lo hemos hecho



Y así es como se muestra el resultado de las sentencias programadas en consola



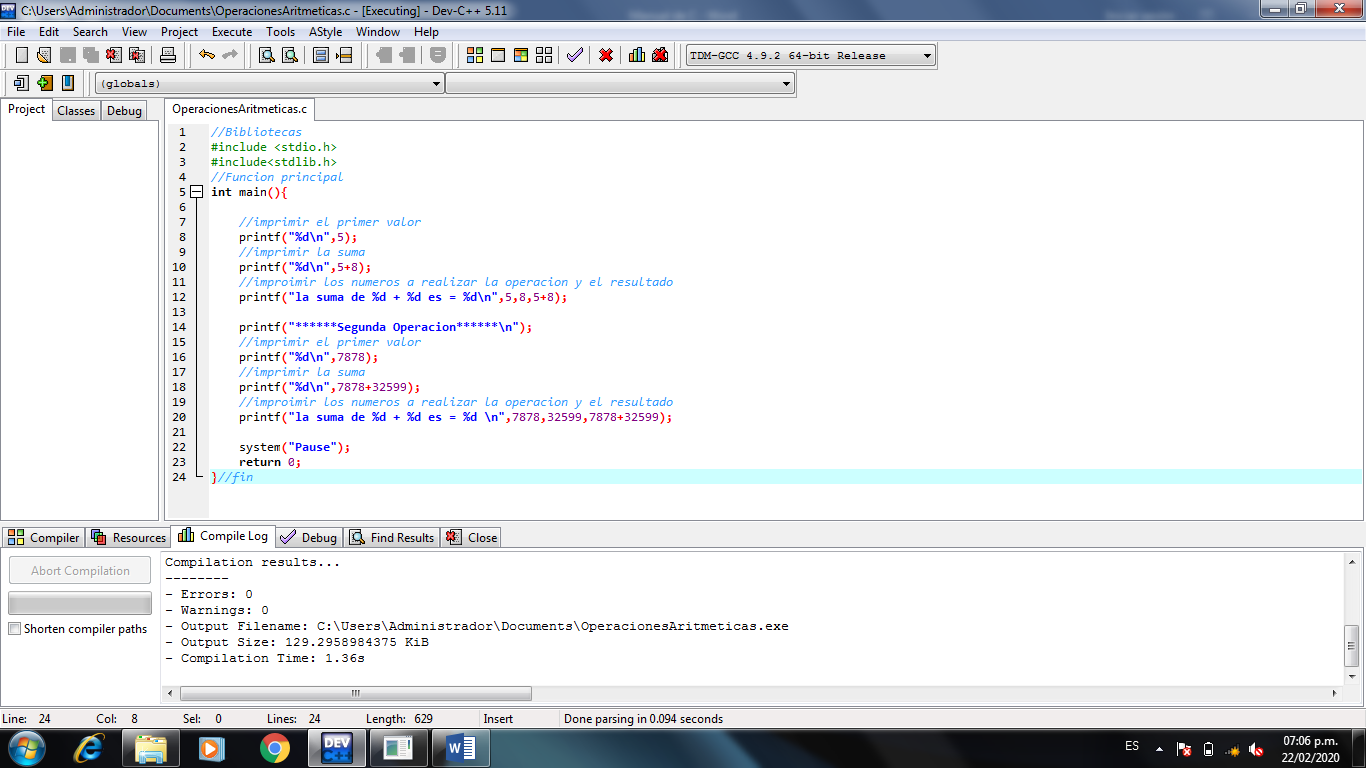
EJERCICIO

Crear un nuevo archivo en Dev-C++ que se llame OpreracionesAritmeticas

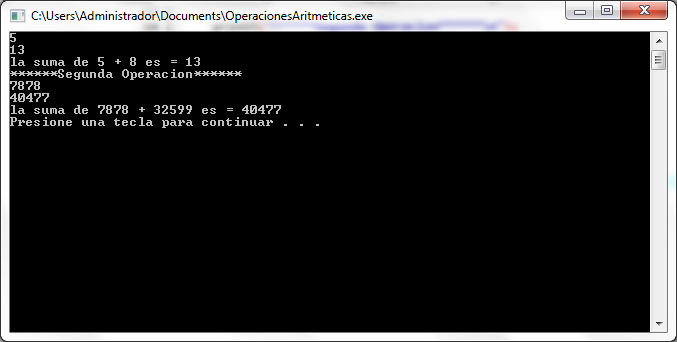
En el método principal (main) escribir el código para escribir la suma de:

* 5+8
* 7878+32599

Por lo tanto, creamos un archivo, al cual guardamos con el nombre anteriormente indicado, agregando así mismo la extensión .c, y una vez dentro del método ´principal, programamos las siguientes sentencias, para dar solución al ejercicio.



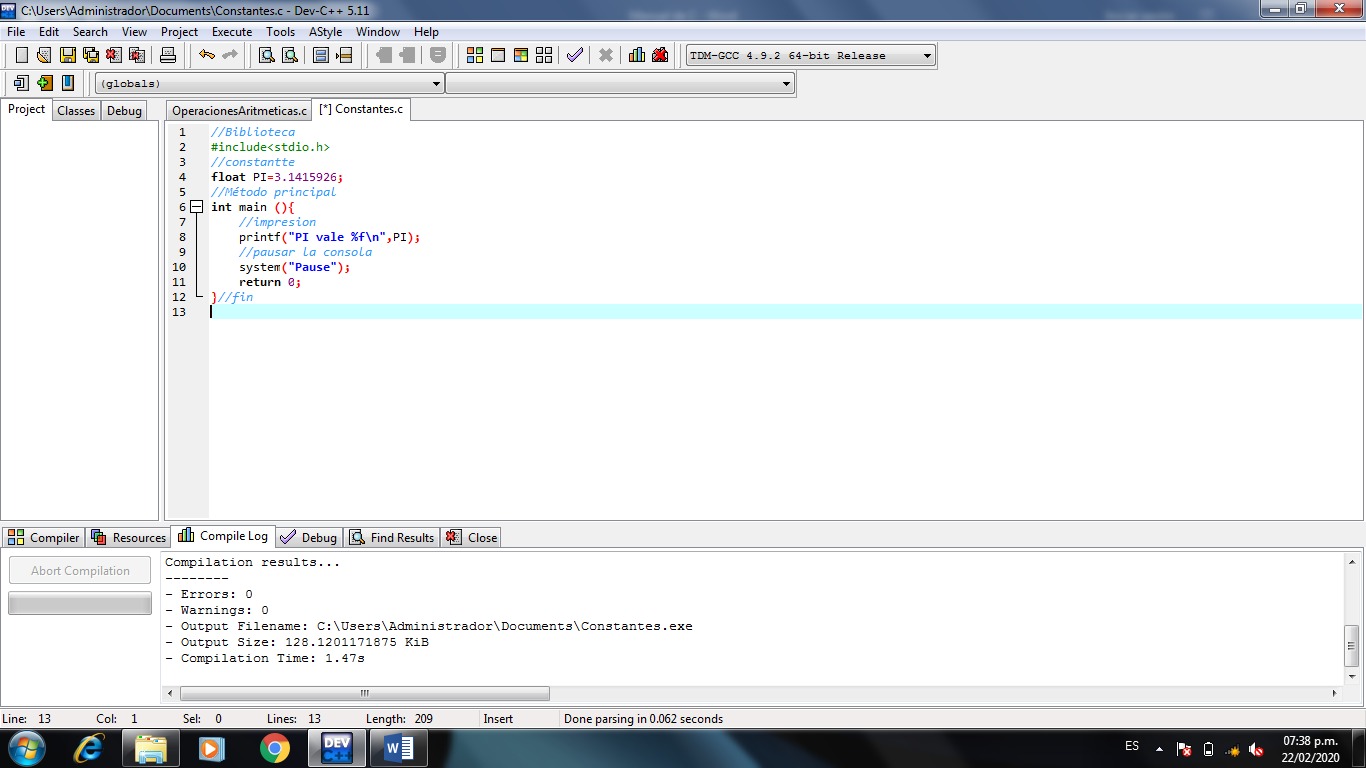
Mientras que al ejecutarse el programa se muestran los siguientes resultados en consola

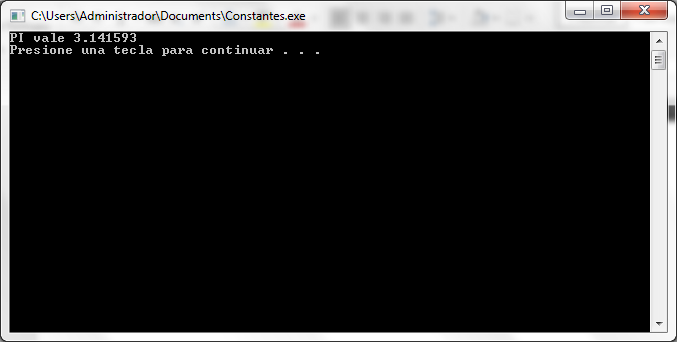


## Lamina3 Variables

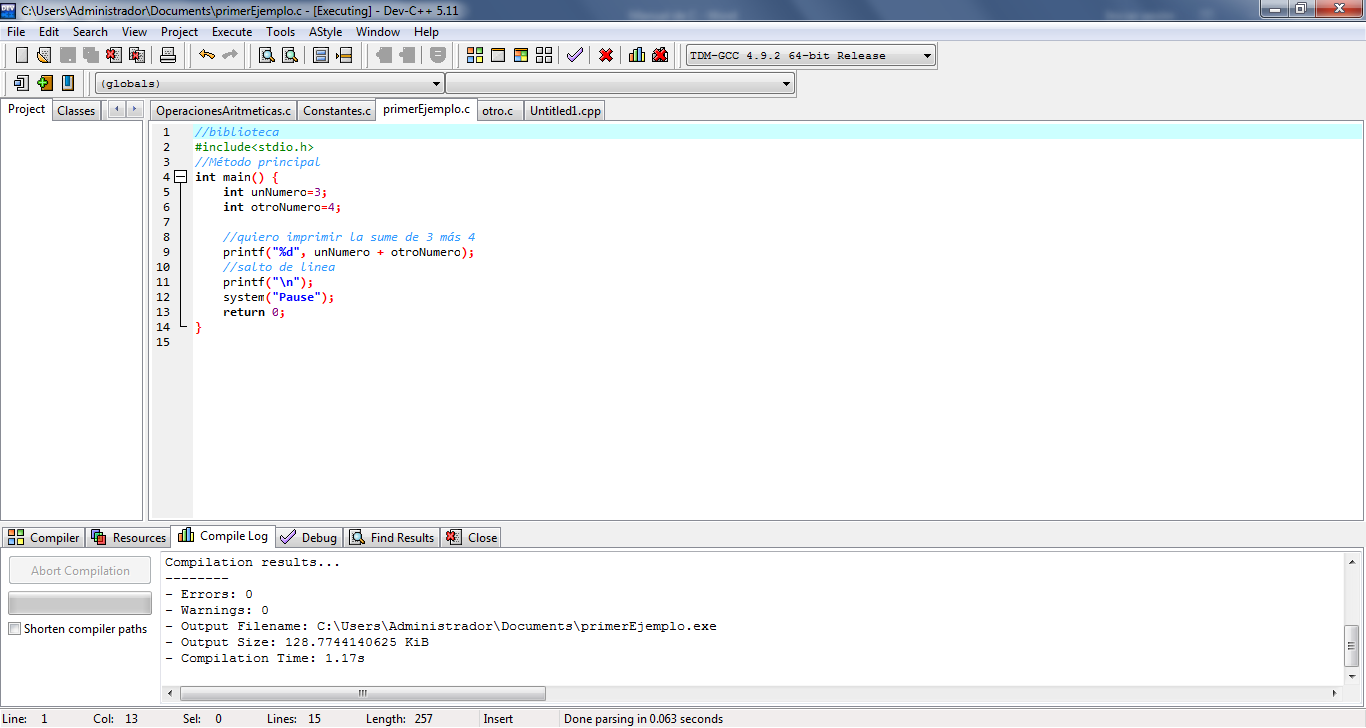
En el siguiente programa se muestra como se utilizan las constantes, los cuales son un espacio en memoria al igual que las variables, sin embargo, en su caso el valor que almacenan no cambia durante la ejecución del programa, para definirlas o declararlas en lenguaje C se hace de forma muy distinta a una variable, por ejemplo



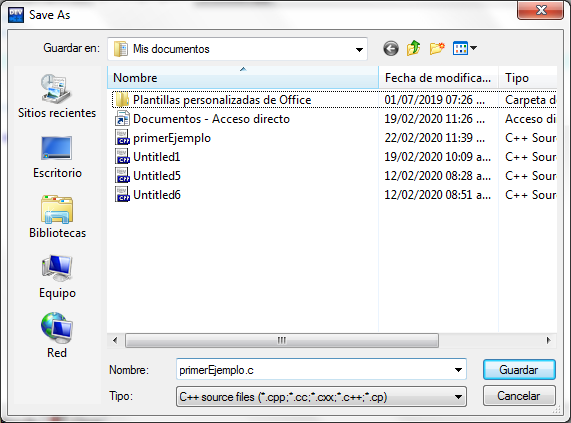


}

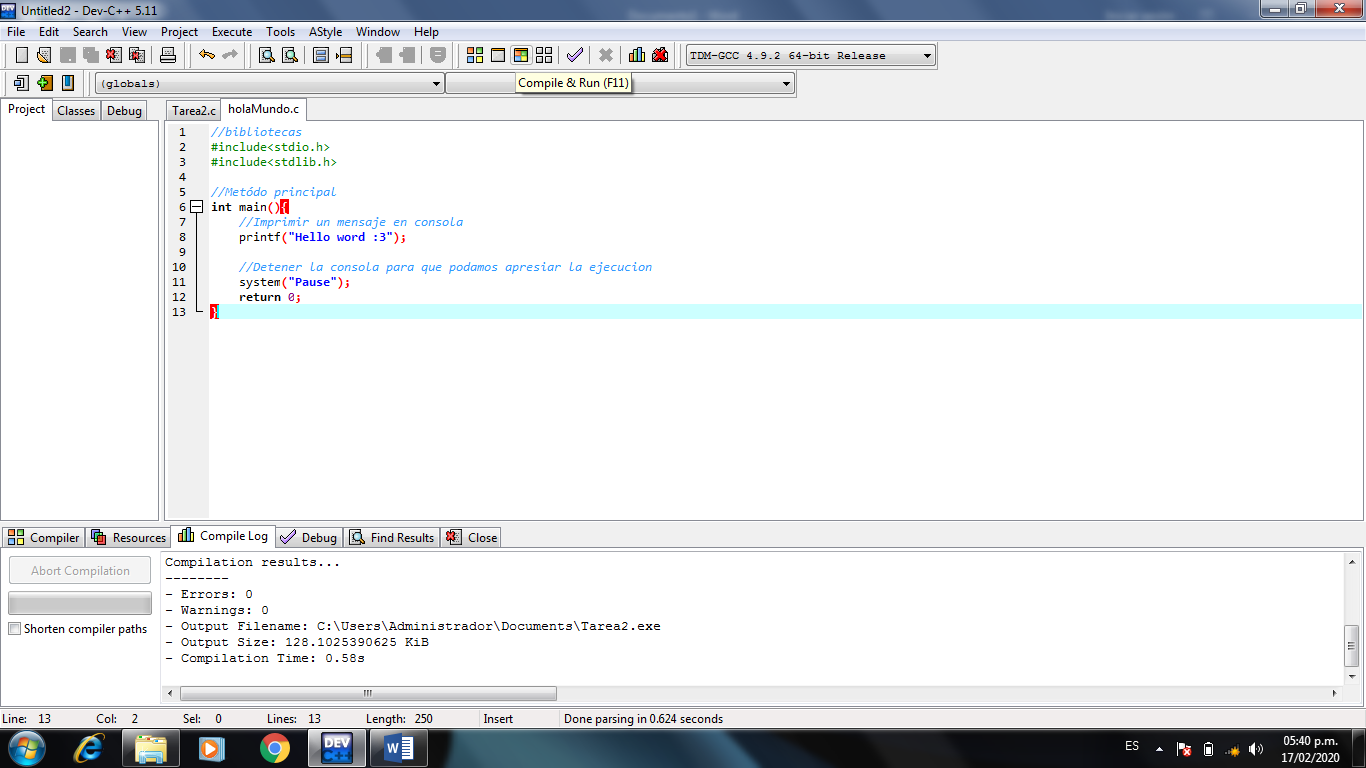
En este segundo ejemplo lo que se realiza es una suma de dos valores de tipo entero, en el cual podemos imprimir el resultado de la sume de ambos sin necesidad de agregar una variable a la cual se le asigne el valor de la suma de dichos números, por lo tanto, abrir un nuevo proyecto en el cual desarrollare el siguiente código.



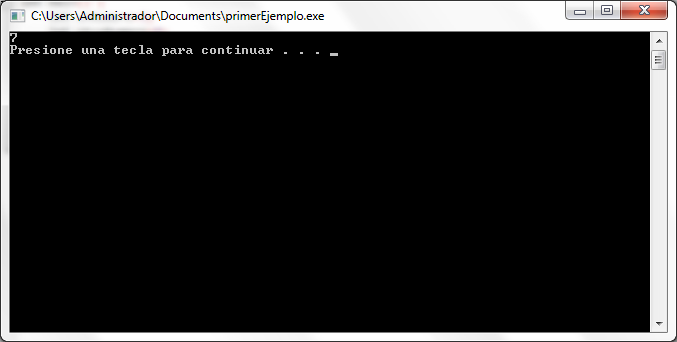
Una vez desarrollado guardare el proyecto con la extensión .c como se muestra a continuación.



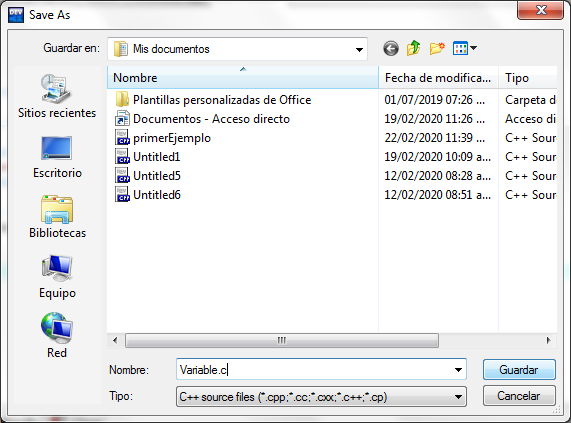
Y ejecuto mi programa dado clic sobre el siguiente icono.



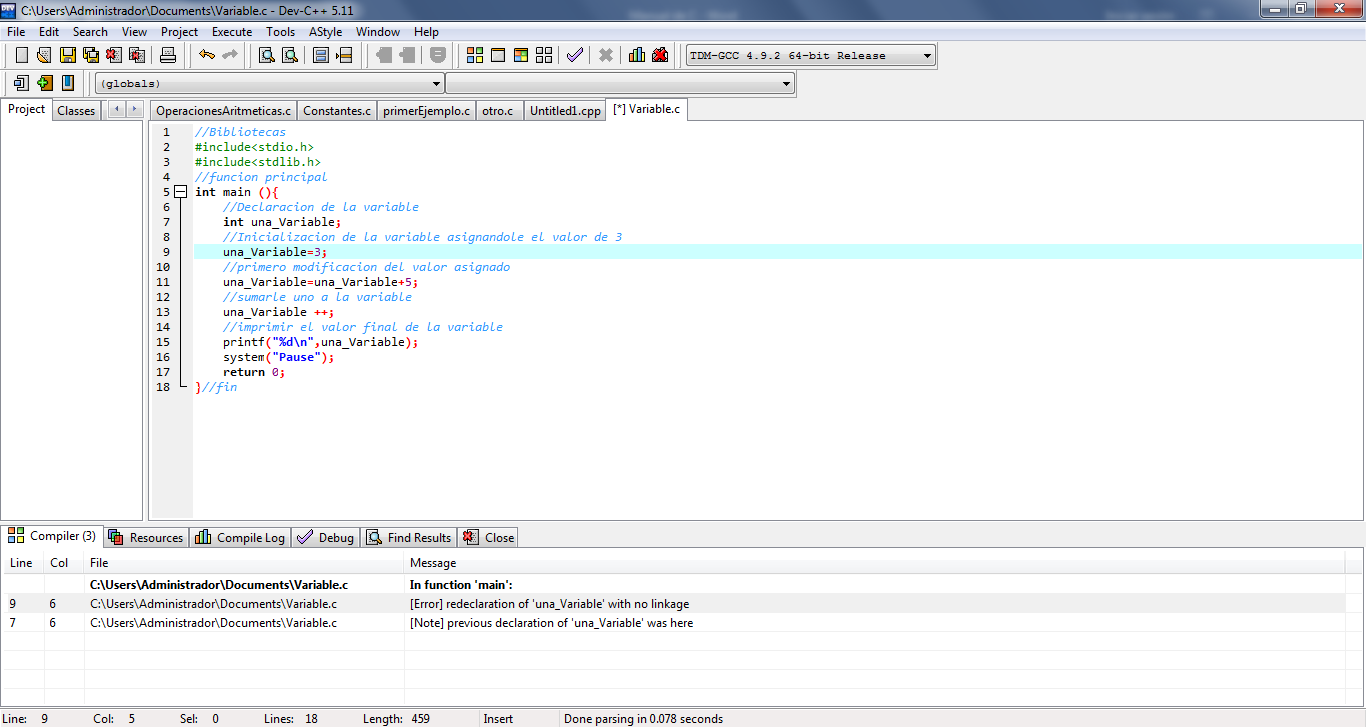
Este es el resultado que se muestra en consola al ejecutar y compilar



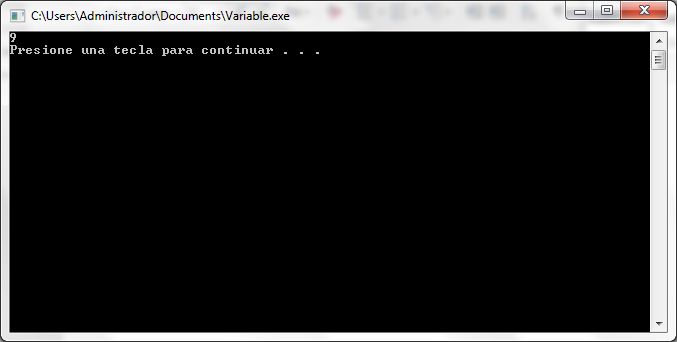
Se crea un nuevo proyecto el cual guardaremos con el nombre de Variables.c



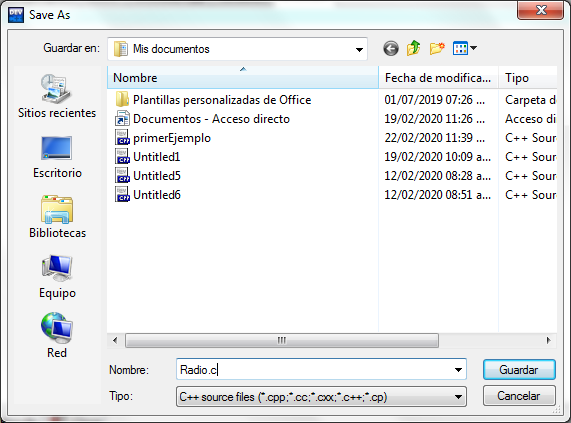
En el cual desarrolle el siguiente código, el cual muestra el comportamiento de una variable durante la ejecución del programa en este caso la variable primero vale 3, posteriormente la variable es igual a si misma más 5, por lo tanto, el valor de la variable ahora seria 8, pero posteriormente a esta variable le hace un incremento, por lo tanto, el valor final de la variable debe ser 9



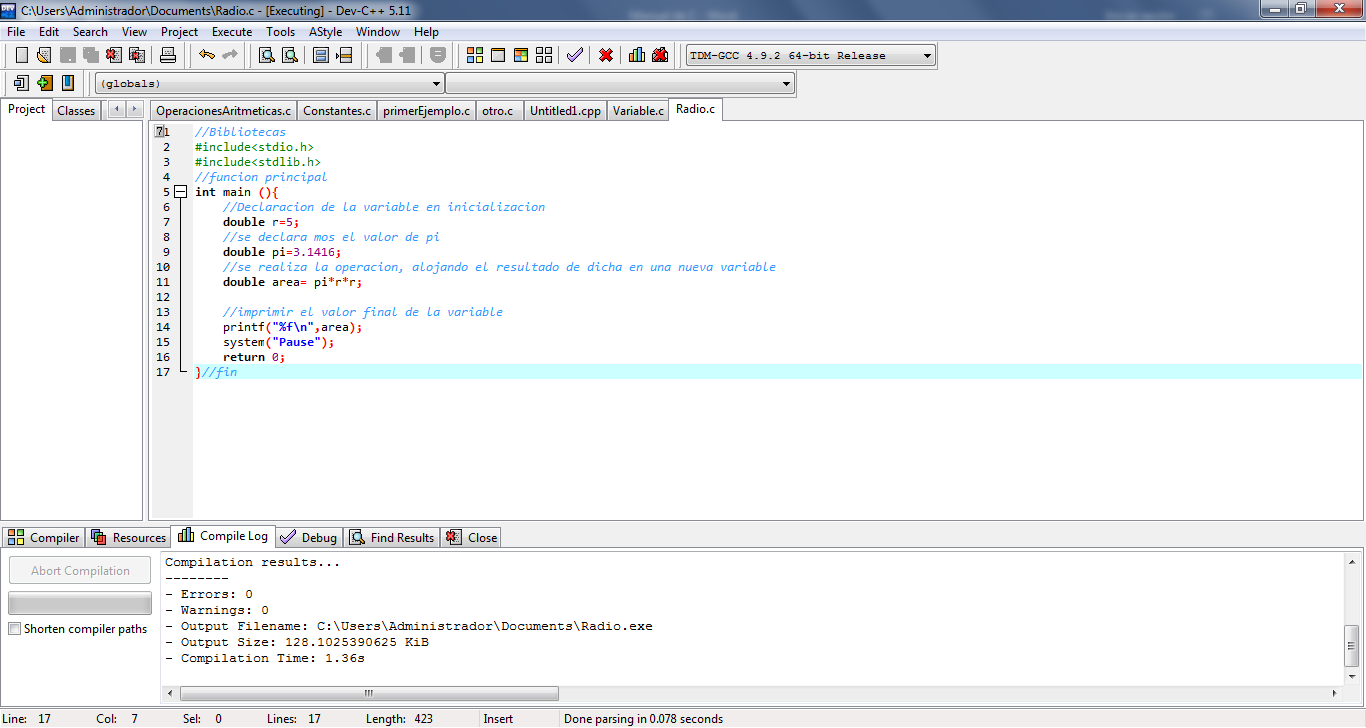
Este es el resultado como anteriormente se mencionó, por lo tanto, se ejemplifica claramente la función de las variables en un programa, ya que modifica su valor por medio de instrucciones.



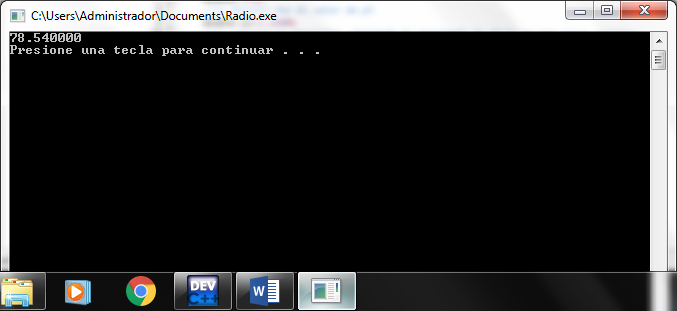
Para obtener el área de un circulo con un radio de 5, se calcula el área, por lo tanto, se crear un nuevo proyecto como anteriormente se ha realizado y guardarlo con la extensión .C, como muestra



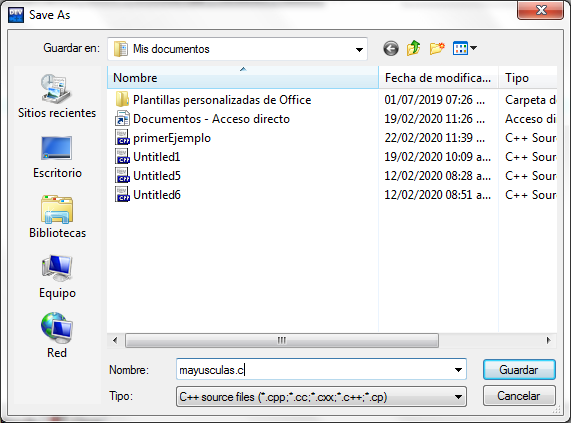
Una vez creado en el área de trabajo se escriben las siguientes instrucciones para obtener el área de un circulo por medio de la formula área= π \* r\*r



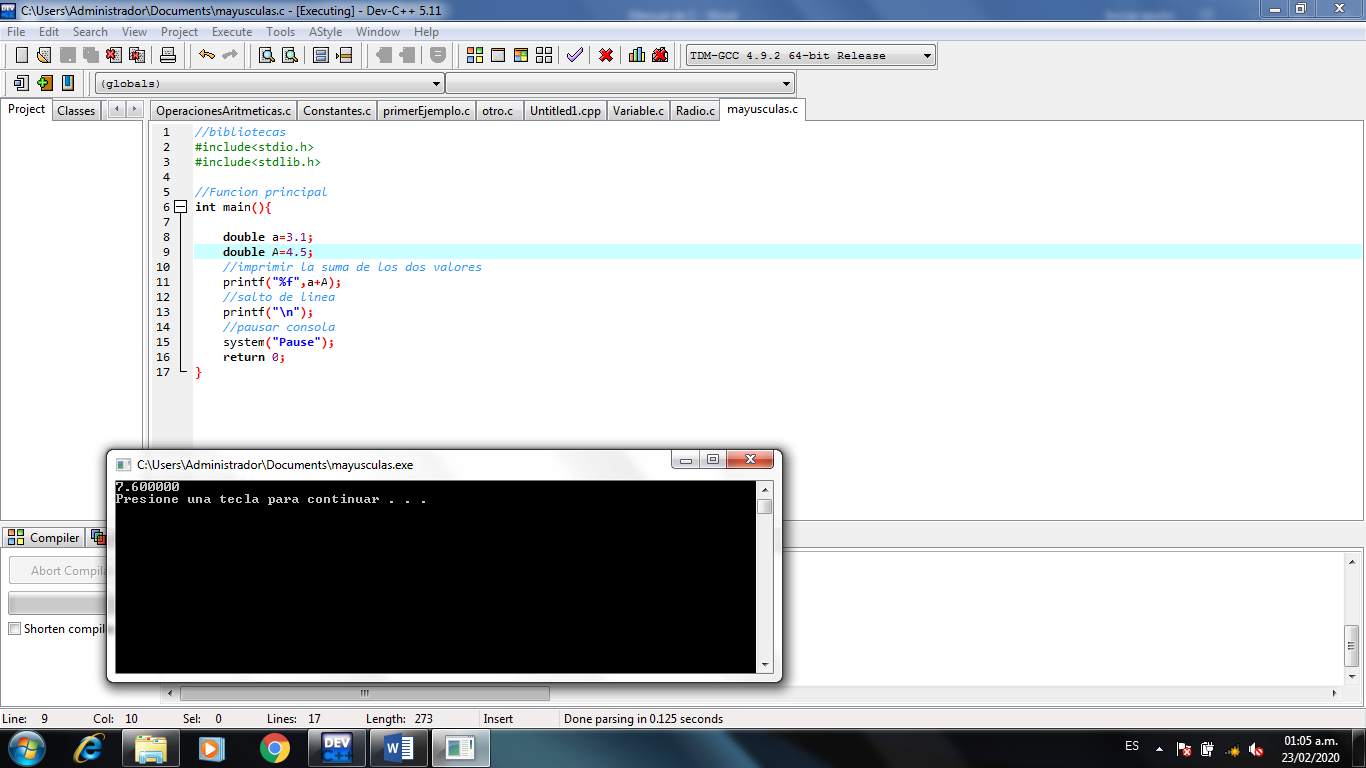
Ejecutamos el programa como anteriormente lo hemos visto



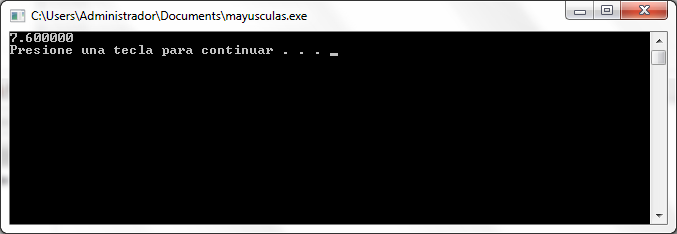
Ahora creo un programa como anteriormente lo he realizado, el cual guardaremos como mayúsculas.c



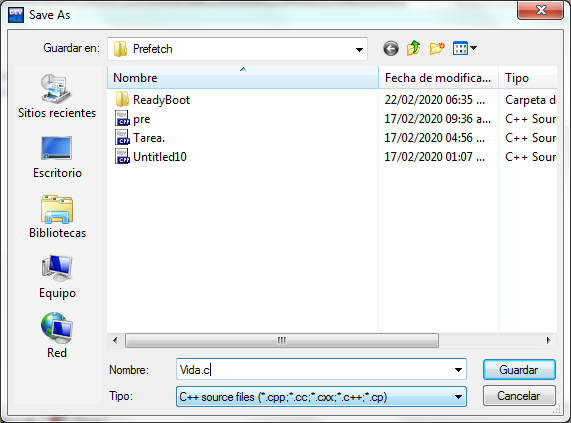
El código que se presenta se elaboró con la finalidad de demostrar que, a pesar de ser el mismo carácter, hay una diferencia si es mayúscula o minúscula, por lo tanto, si acepta la asignación de valores diferentes



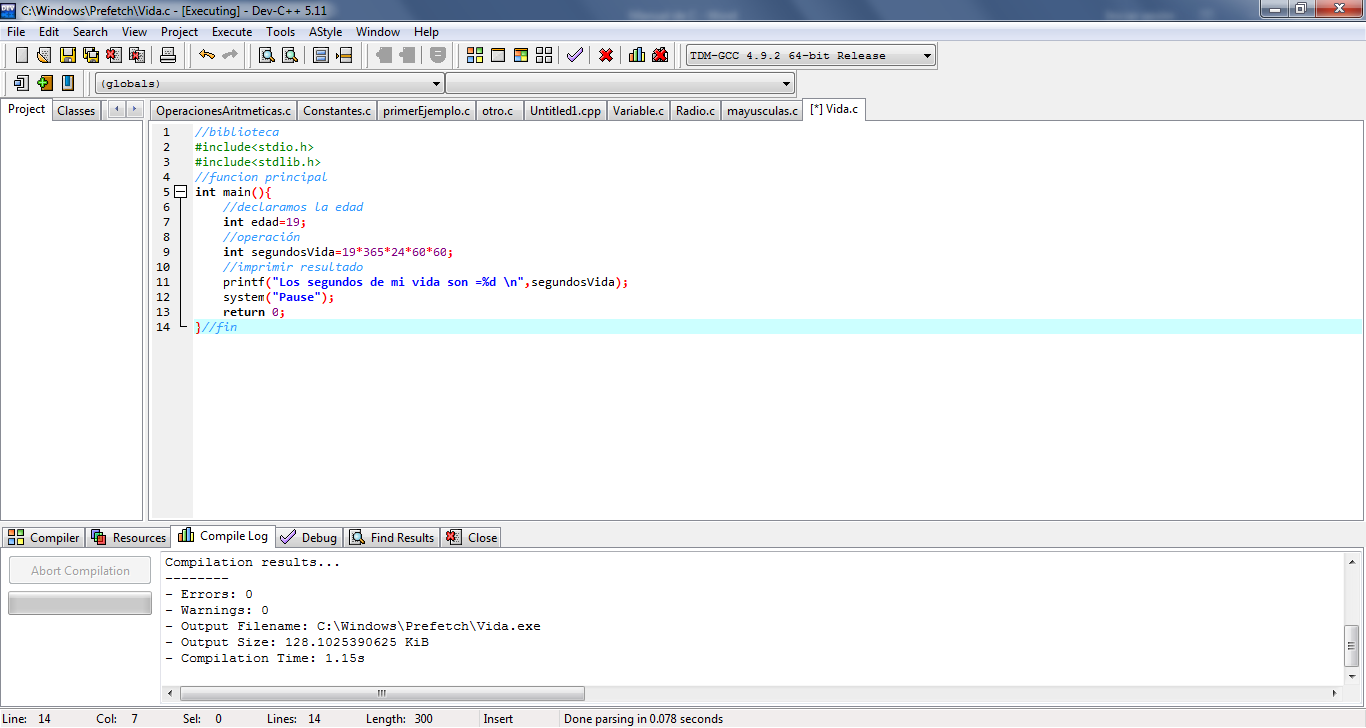
Por ello el resultado de los valores sumados es el siguiente al ejecutarlo en consola



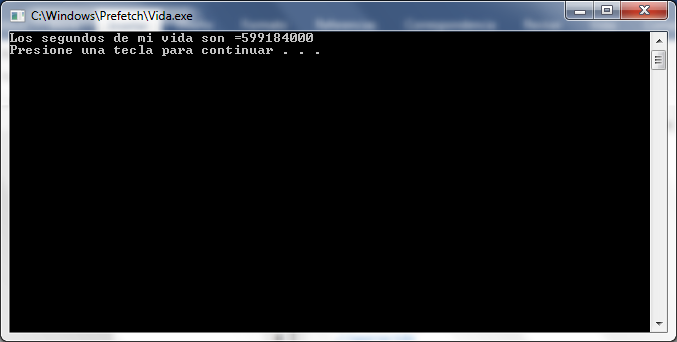
Abrir un nuevo proyecto el cual llame vida.c ya que hace alusión a la función que tendrá la cual será “Calcular la cantidad de segundos que has vivido”.



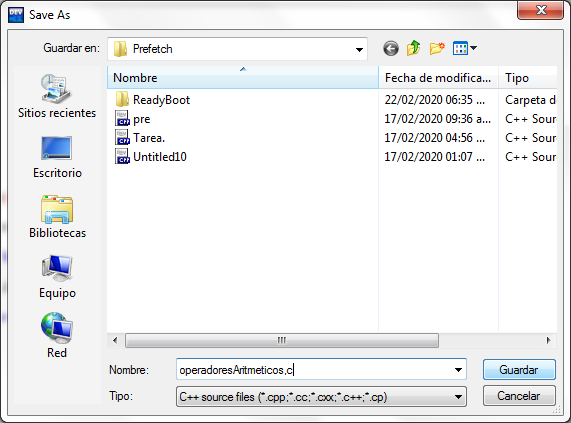
En nuestro código una vez que se estableció la edad se multiplica por 365(días del año), por 24(horas de un día), por ello 60(minutos de una hora), , por 60(segundos de un minuto).



Finalmente se ejecuta y da como resultado el siguiente.

s

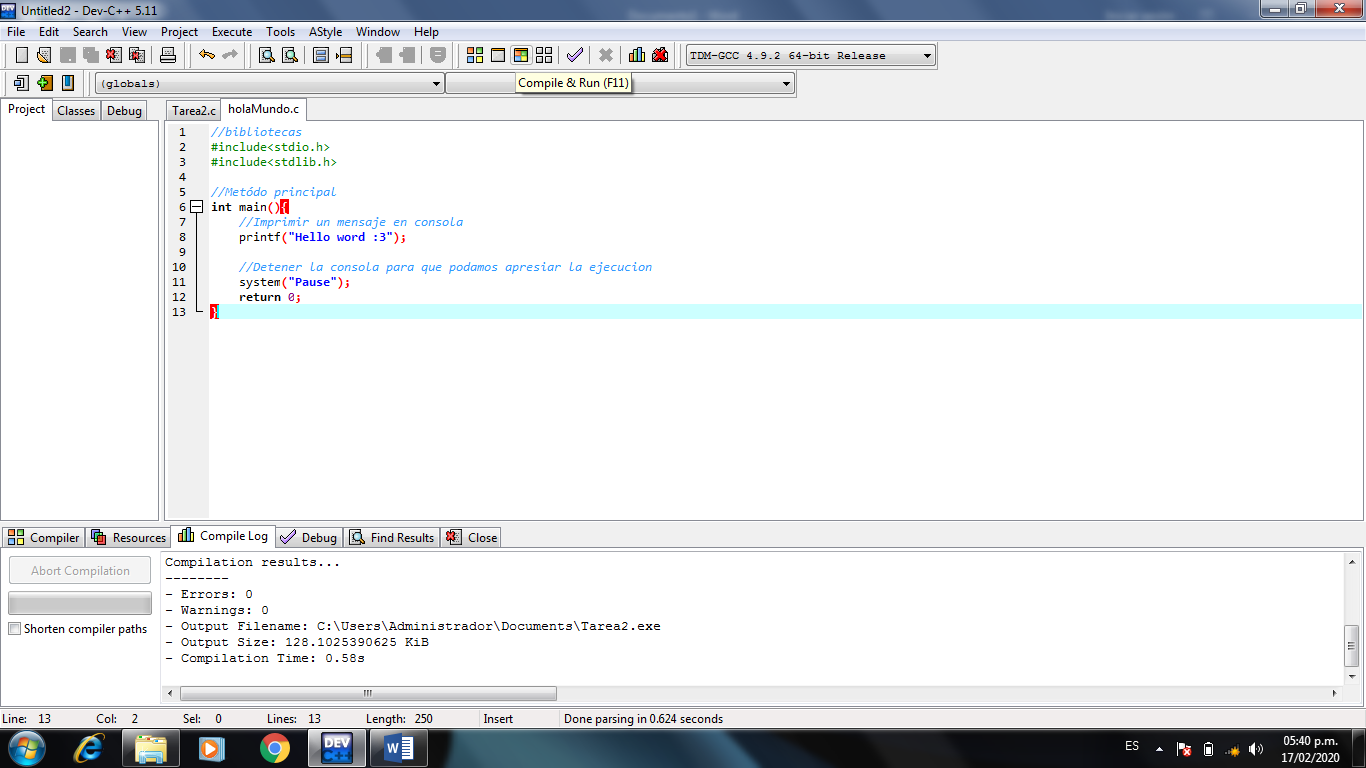
A continuación, creamos una práctica para demostrar como funcionan y se aplican los operadores aritméticos, mostrando cada un de las operaciones básicas y fundamentales, para ello creamos un muevo proyecto el cual llamamos operadoresAritmeticos.c, como se puede observar.



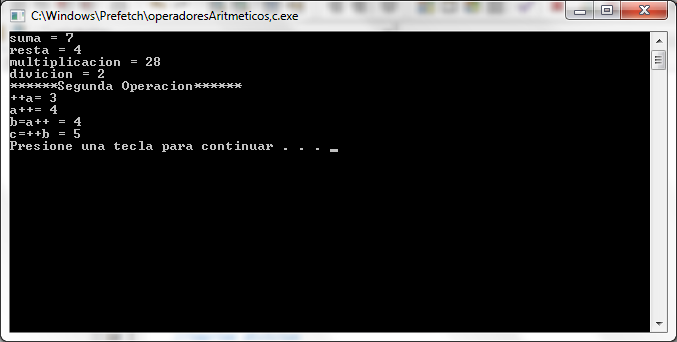
Una vez en el área de trabajo, declaramos las operaciones fundamentales y as imprimimos posteriormente para apreciar como funcionan.



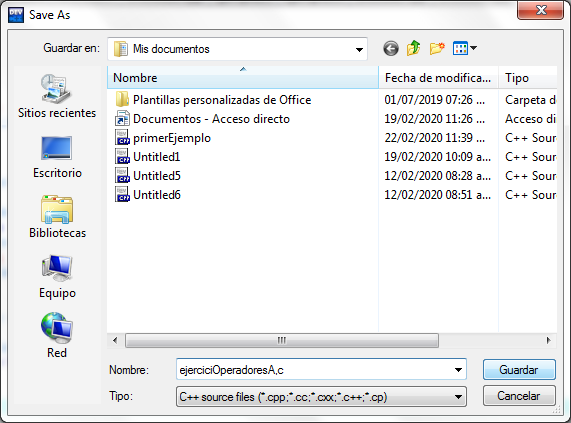
A continuación, ejecutamos nuestro programa dando clic en el siguiente icono



Y se muestra así nuestras impresiones en consola y cómo podemos apreciar que es muy diferente el pre decremento del decremento.



Esta práctica se demuestra la función de los operadores aritméticos de multiplicación y división, ya sea para números reales o enteros según corresponda, por lo tanto, creamos nuestro proyecto llamándolo como se muestra.



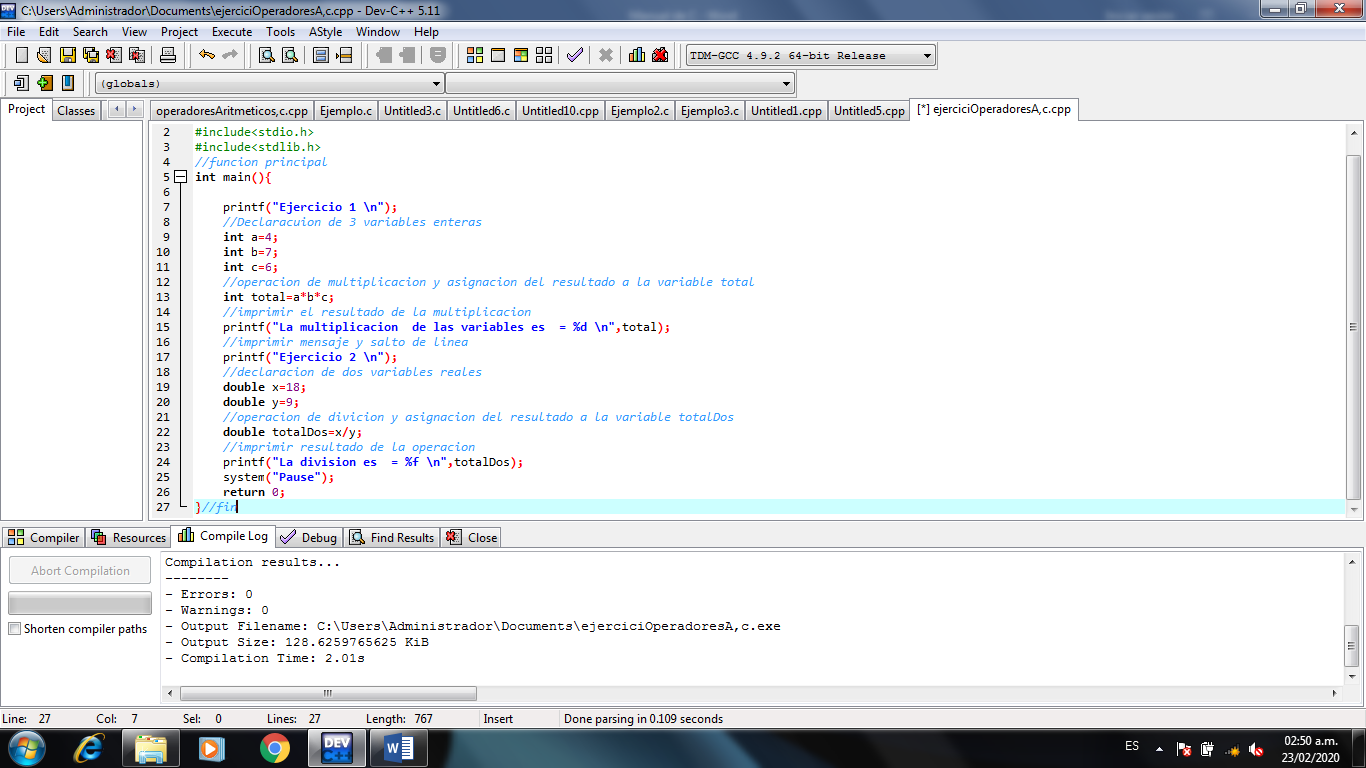
El ejercicio consta de

Ejercicio1

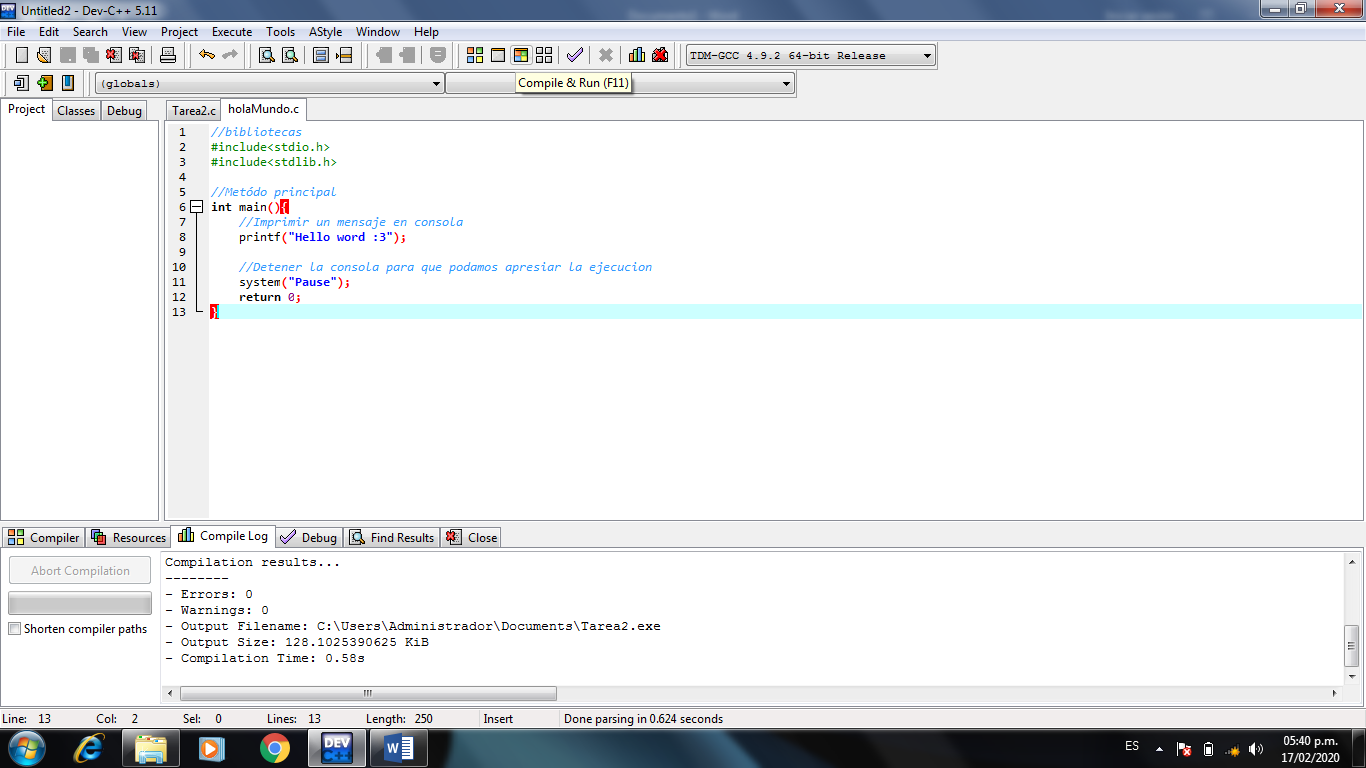
* Declarar tres variables y asignarles valores enteros.
* Mostrar el resultado de la multiplicación de las tres variables

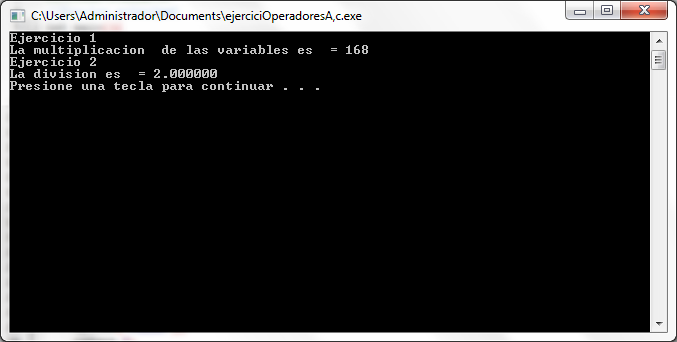
Ejercicio2

* Declarar dos variables dobles(reales) y asignarles valores diferentes de 0.
* Mostrar el resultado de dividir la primera entre la segunda.

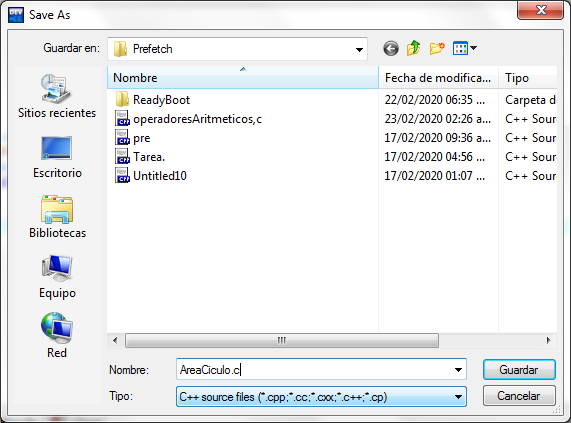


Ejecute y se muestran los mensajes programados en consola.



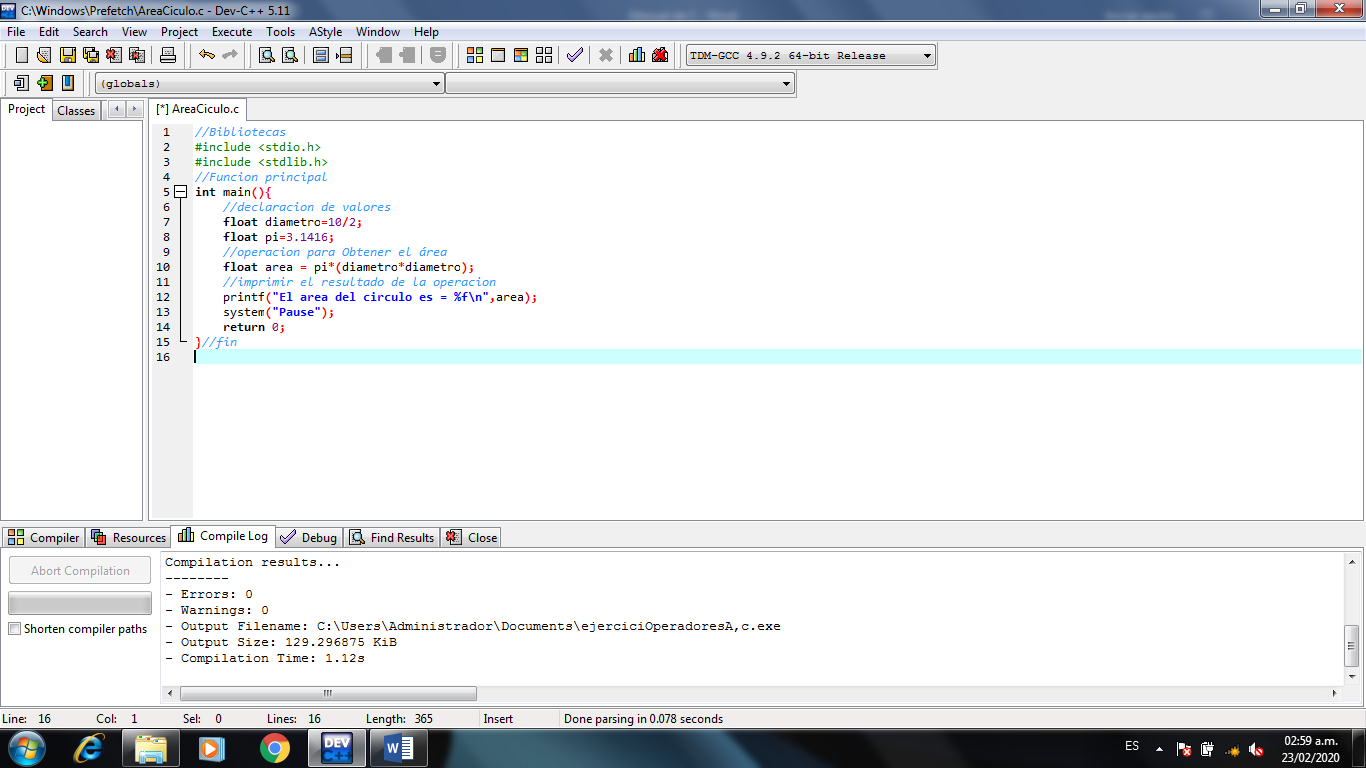


Abrimos nuevamente un programa que nos permite calcular el are de un circulo, por ello lo llame AreaCirculo.c

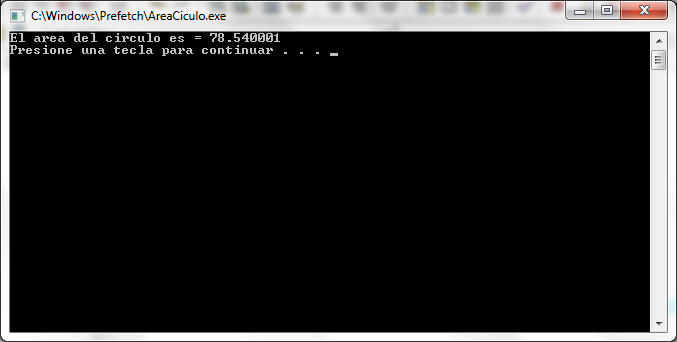


Posteriormente desarrollaos el código para dar solución al siguiente ejercicio

* Crea un programa que calcule el área de un circulo que tiene 10 metros de diámetro

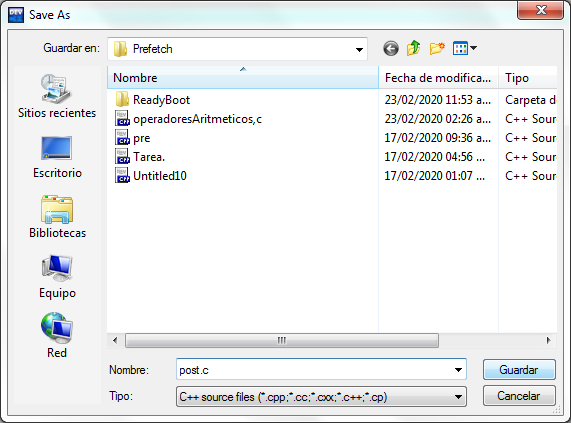


Ejecutamos y se muestra el resultado.



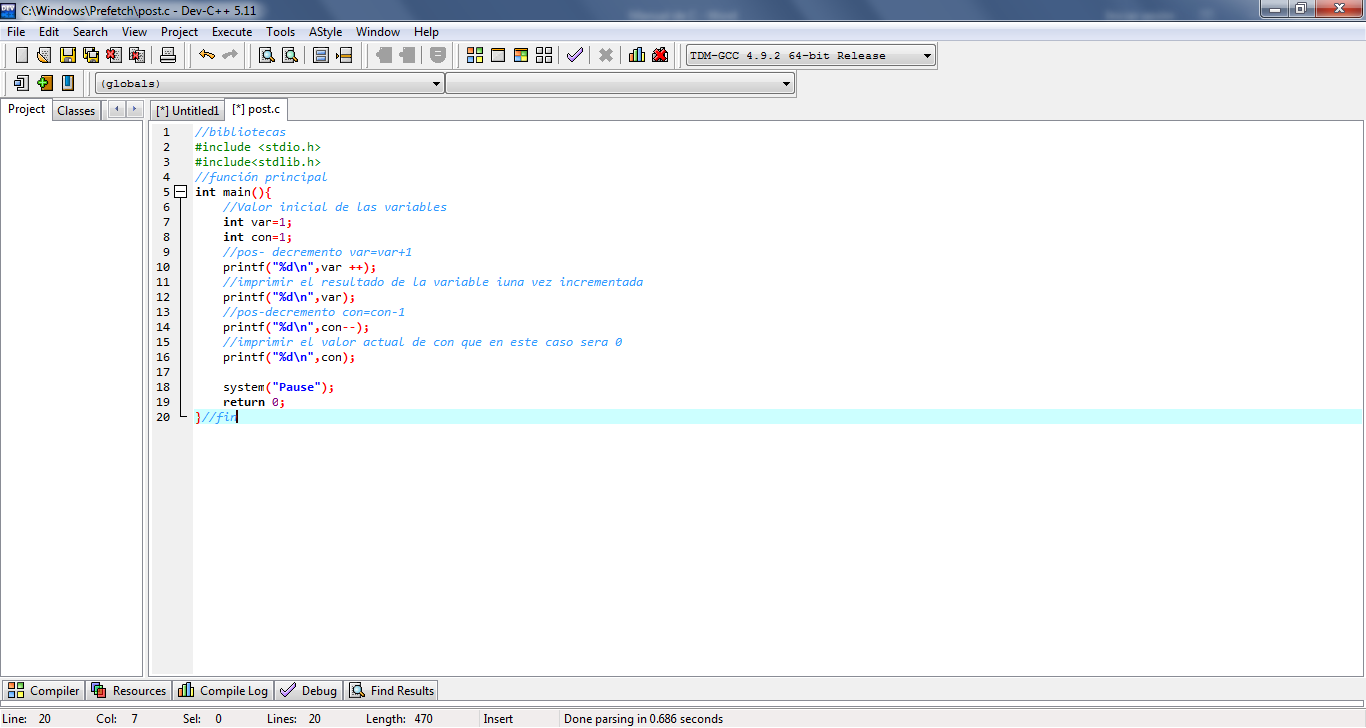
## Lamina 4 Operador Incremento

A continuación, se crea un nuevo proyecto con el nombre de post, como se muestra a continuación

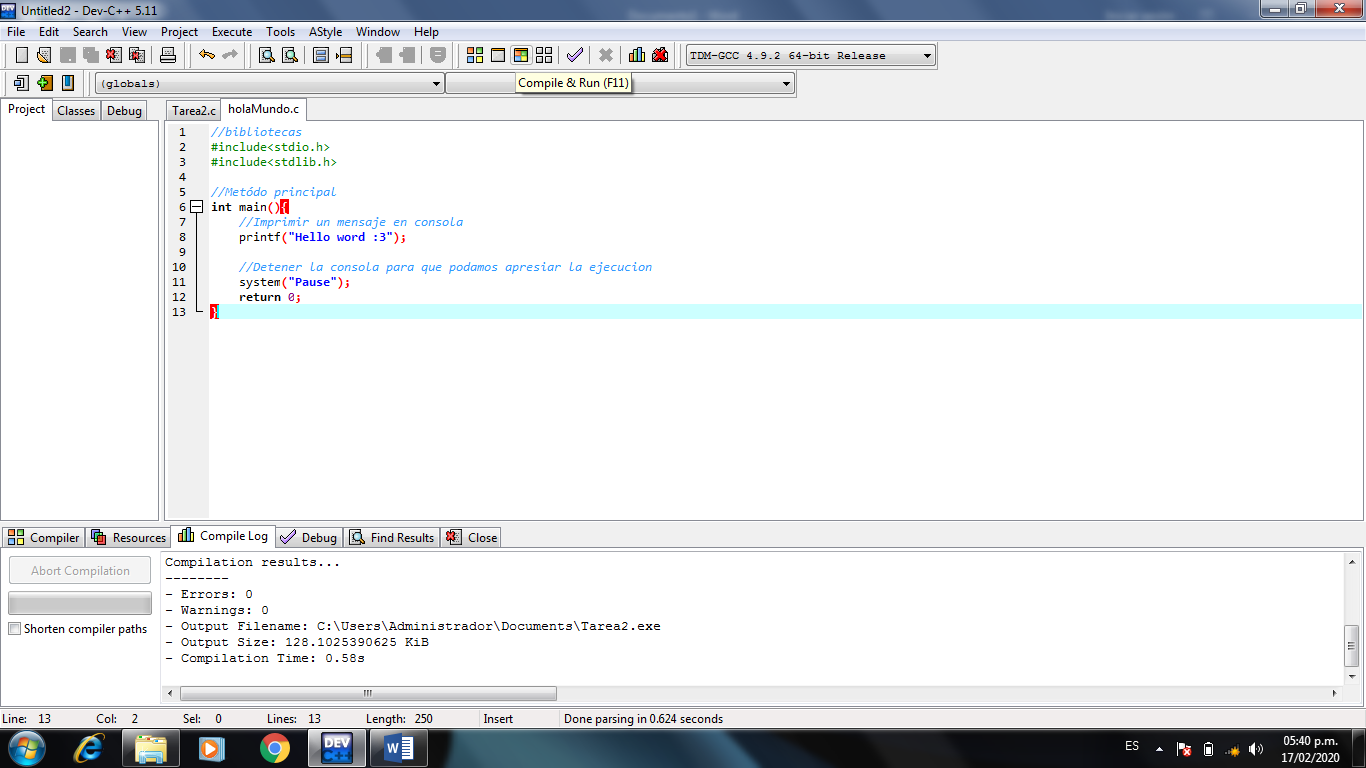


En el cual desarrollamos un código mostrando la funcionalidad del post-incremento o post-decremento, el cual consiste en:

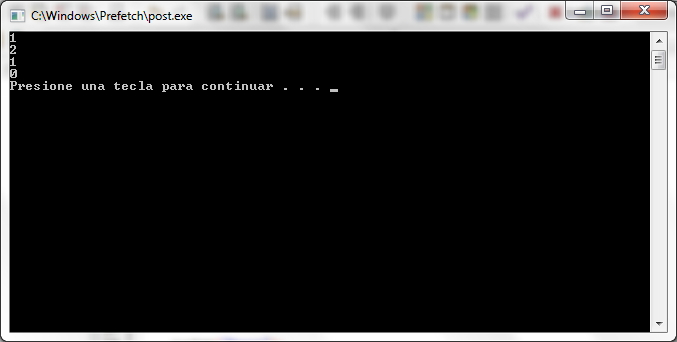
Si tememos una variable a esta se le hace un incremento o decremento a su valor actual, por lo tanto, el valor cambia más uno o menos uno tal sea el caso como lo es en nuestro ejemplo.



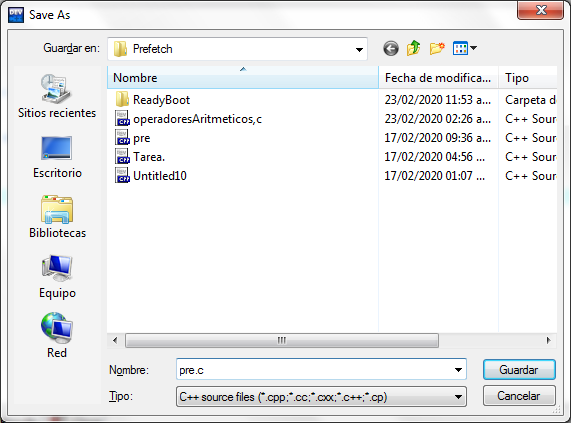
Ejecuto el proyecto



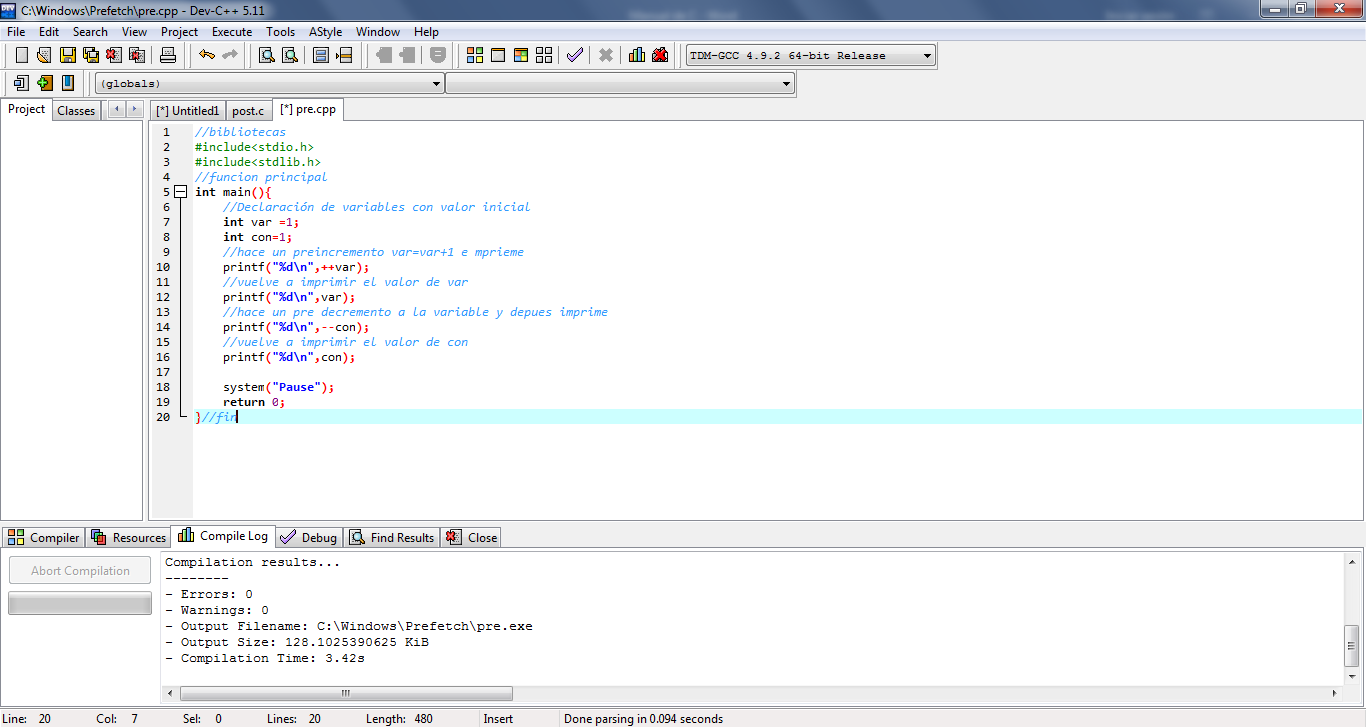
En a impresión se muestra el resultado de las acciones programadas, en el cual la variable var=1 y al incrementarle uno imprime2, mientras que la variable con=1 y al decrementar 1 su valor final es de 0



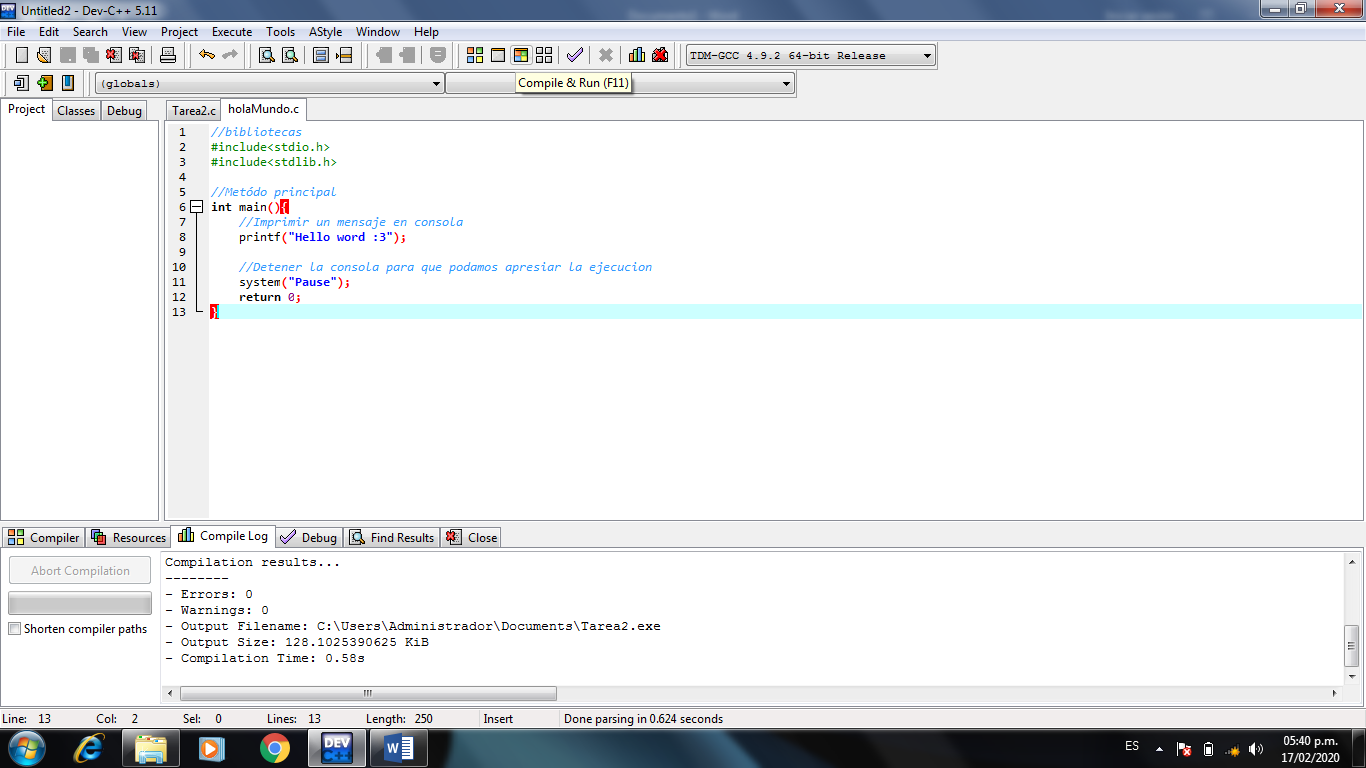
Ahora creamos un nuevo proyecto con el nombre de pre, como se muestra en la captura



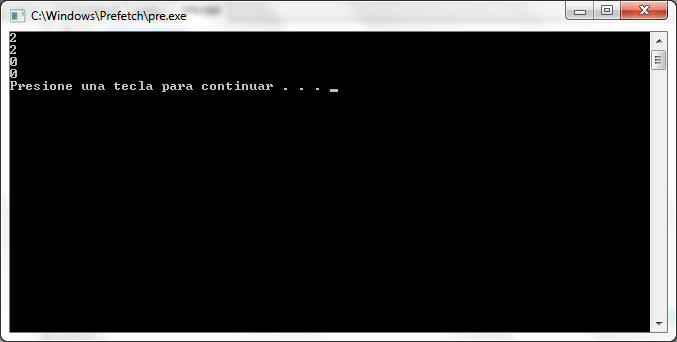
En el área de trabajo desarrollamos el código que consiste en que una variable ya inicializada en ella lo que se hace primero es la operación, es decir primero la incrementa o decremente y después es utilizada.



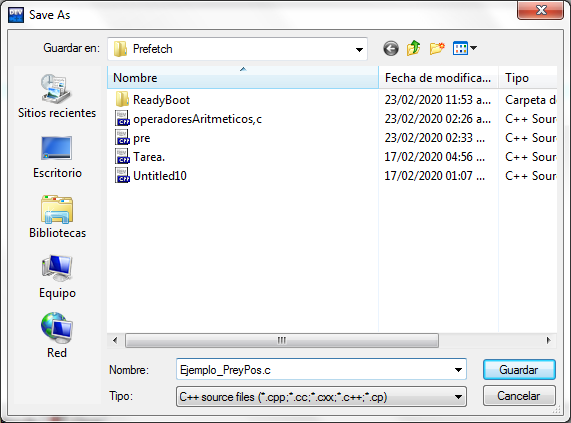
Después se ejecuta el programa



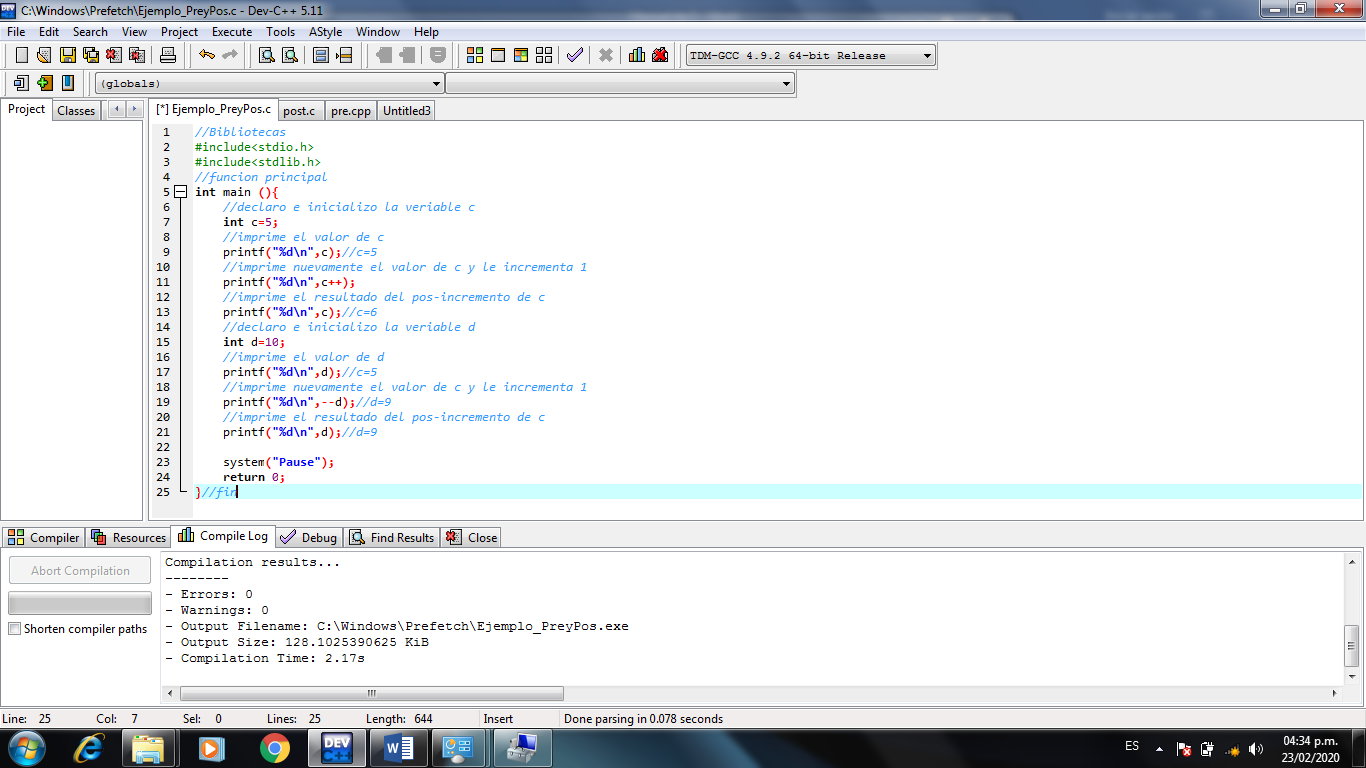
Por ello en el pre-decremento como se utiliza primero será var=1, después se le incremente uno e imprime el valor y en el pre decremento primero se decrementa y después imprime la variable es decir la utiliza.



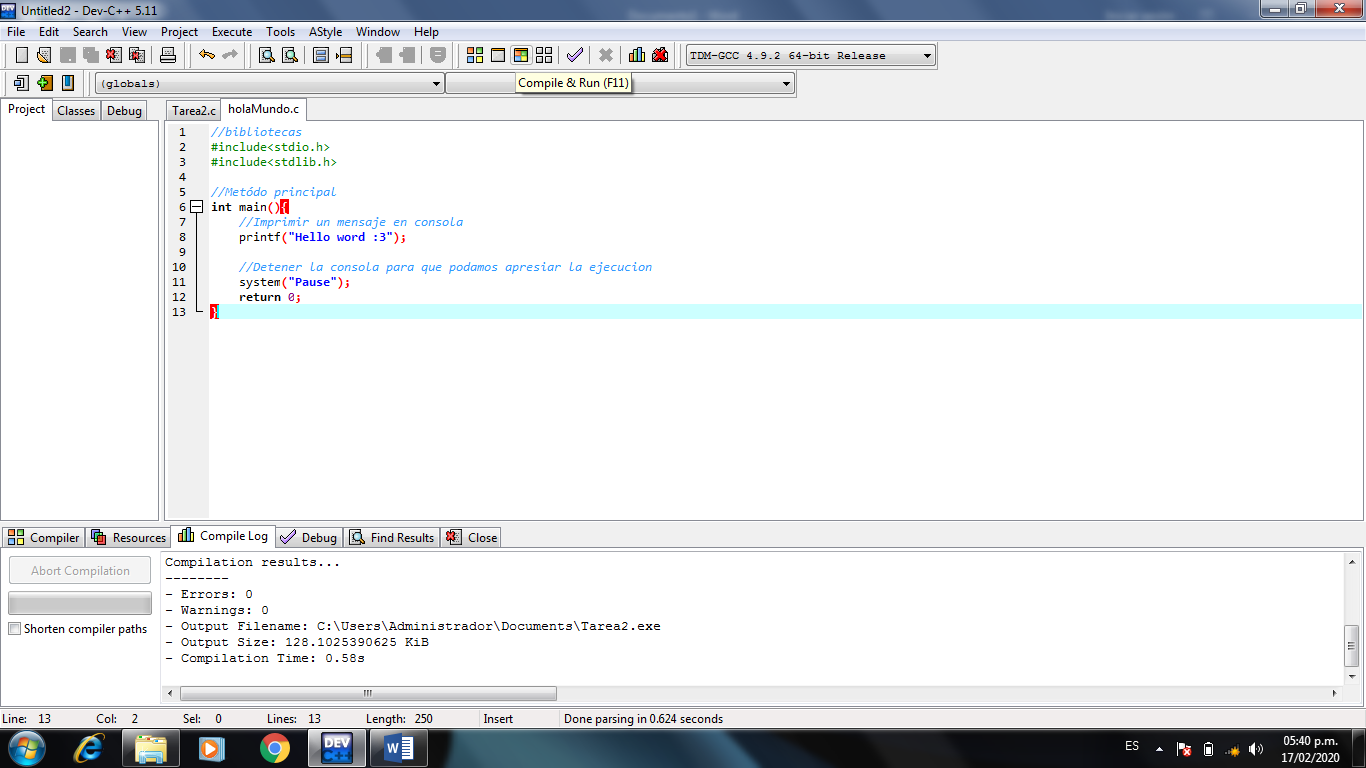
Cree un nuevo proyecto como anteriormente se ha realizado, con la finalidad de analizar más a fondo el uso del pos-incremento y pre-decremento, mientras lo nombrare de la siguiente manera para hacer alusión a la función que tiene.



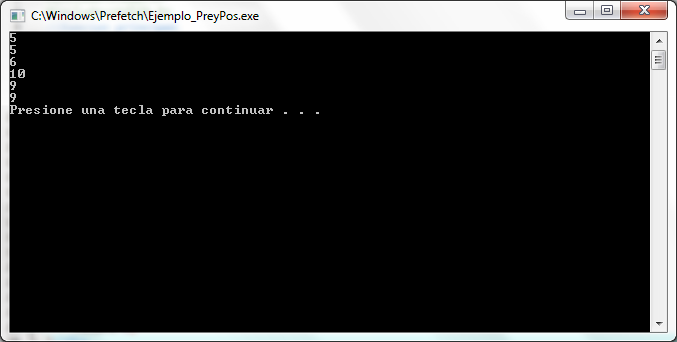
Posteriormente desarrollo el siguiente código, el cual permite contemplar de forma más clara la diferencia entre post y pre



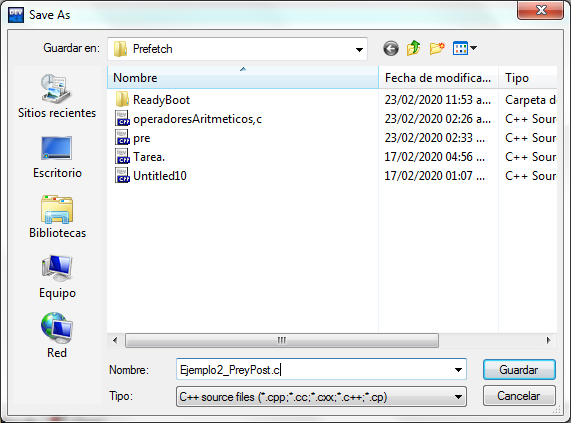
Ejecuto el programa, dando clic sobre el siguiente icono



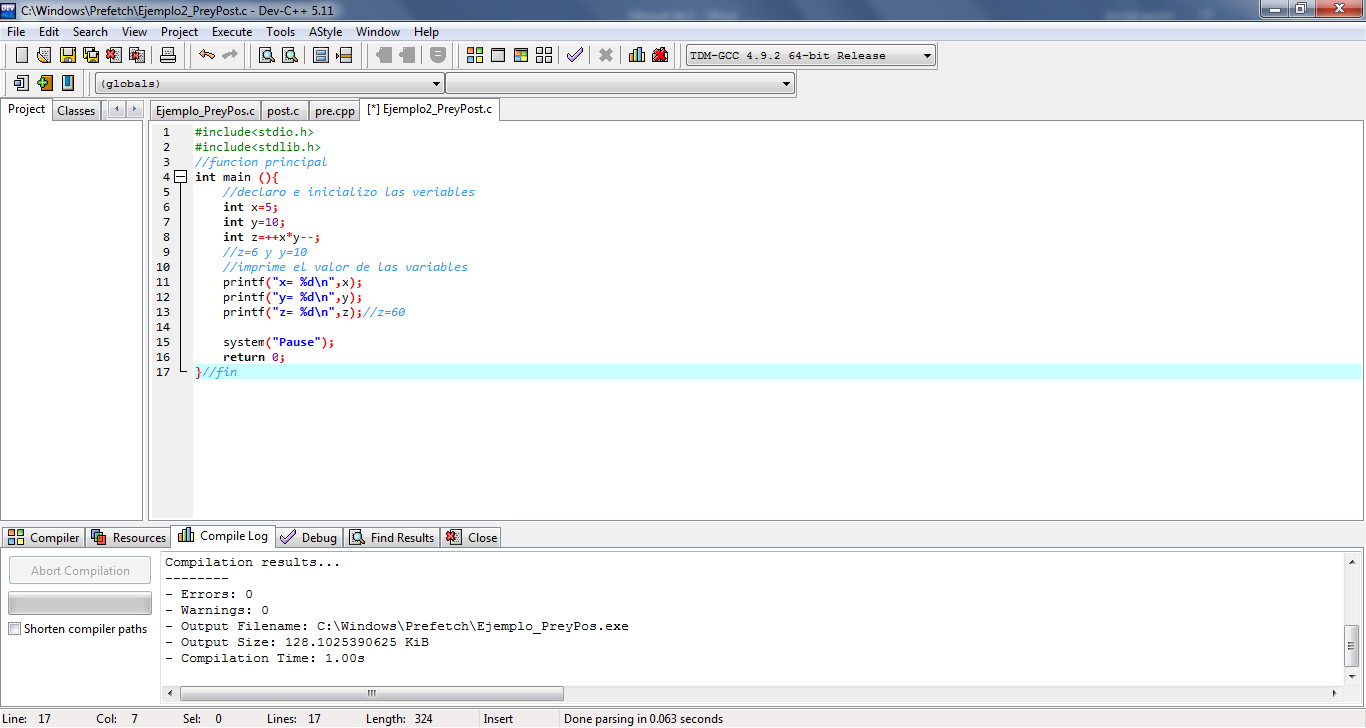
Y este es el resultado obtenido, que como podemos ver es muy distinto, tanto para declarar como su función.



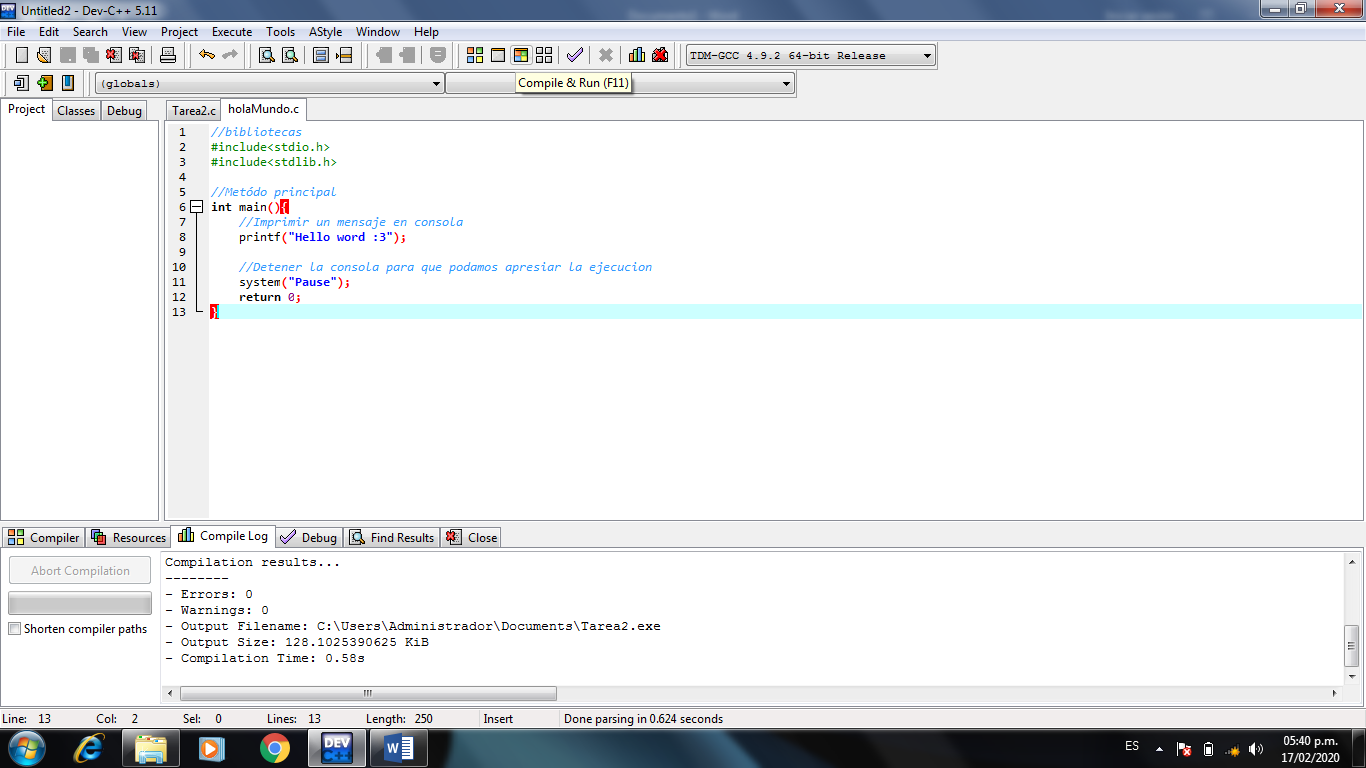
Seguido de ello realizamos una nueva operación, la cual se basa en utilizar el pre-incremento y post-decremento, para iniciar creamos un nuevo proyecto y lo nombramos.



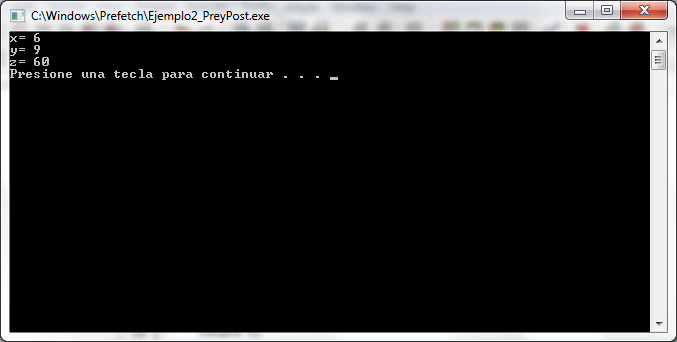
Desarrollamos nuestro código en el cual se muestra que como en x se hace el pre incremento entonces será 6 y después se utiliza la variable para multiplicarla con y, pero como y tiene pos decremento entonces se utiliza el valor de 10 y después la decrementa.



Sucesivamente ejecuto mi programa

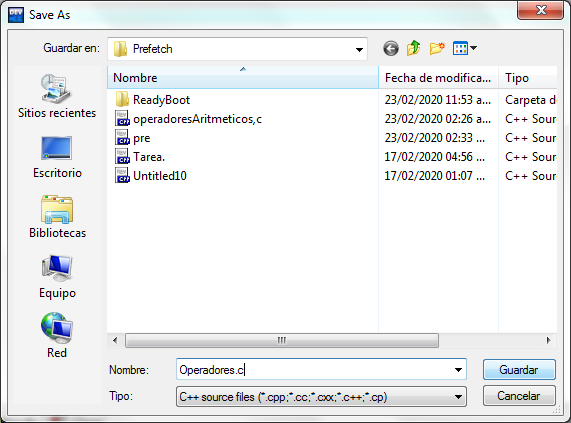


Por lo tanto, el resultado a esperar para z es 60

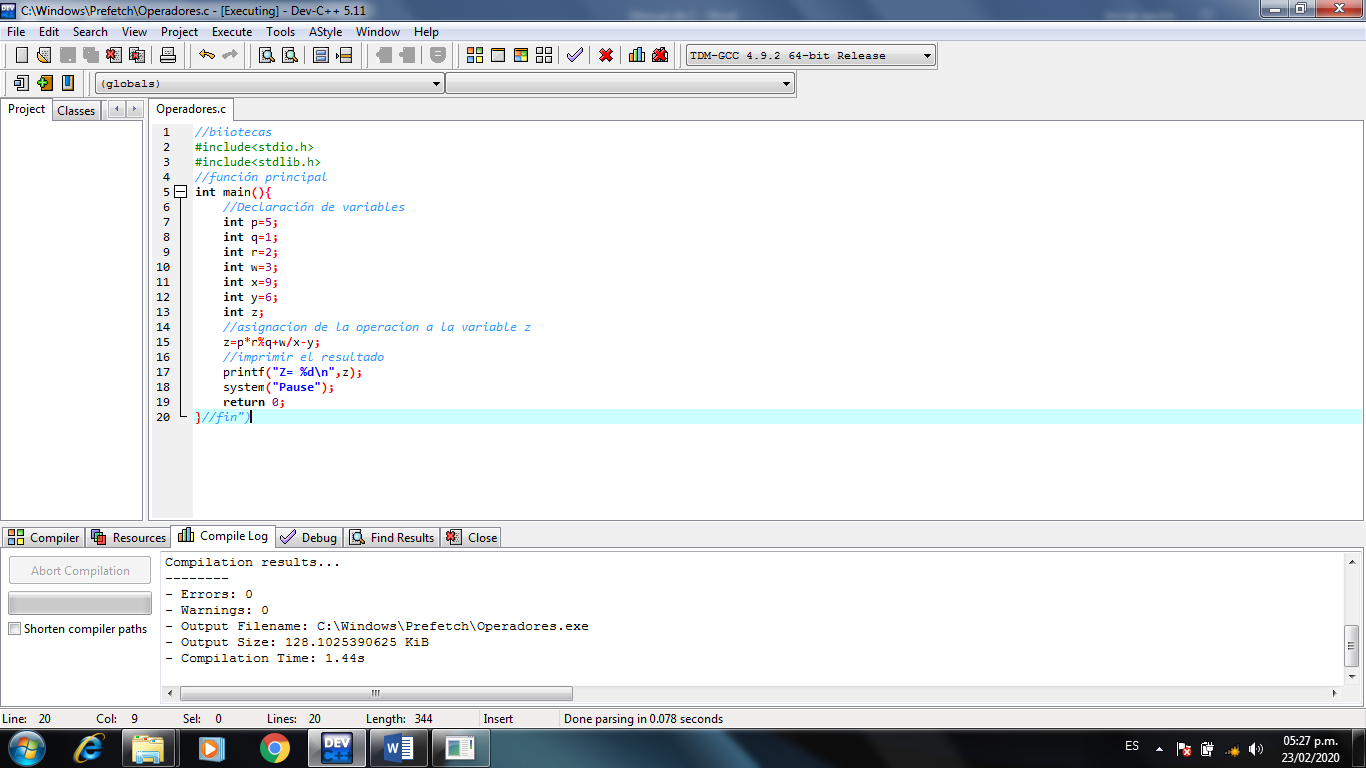


## Lamina 5 Precedencia de operadores

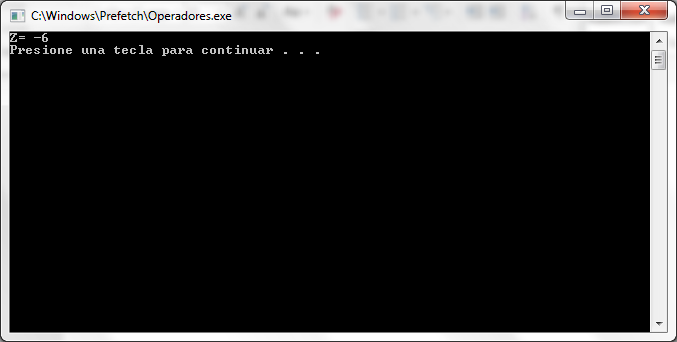
Para comenzar cree un nuevo proyecto llamado operadores.c



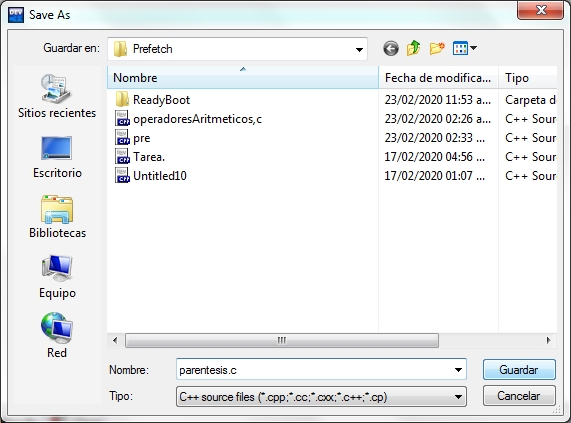
En este código se muestra una operación, sin embargo, la finalidad es saber la prioridad que se da al resolverla, ya que lo primero que se ejecutara es la multiplicación, después el modulo, a continuación, la división, después la suma y finalmente la resta.



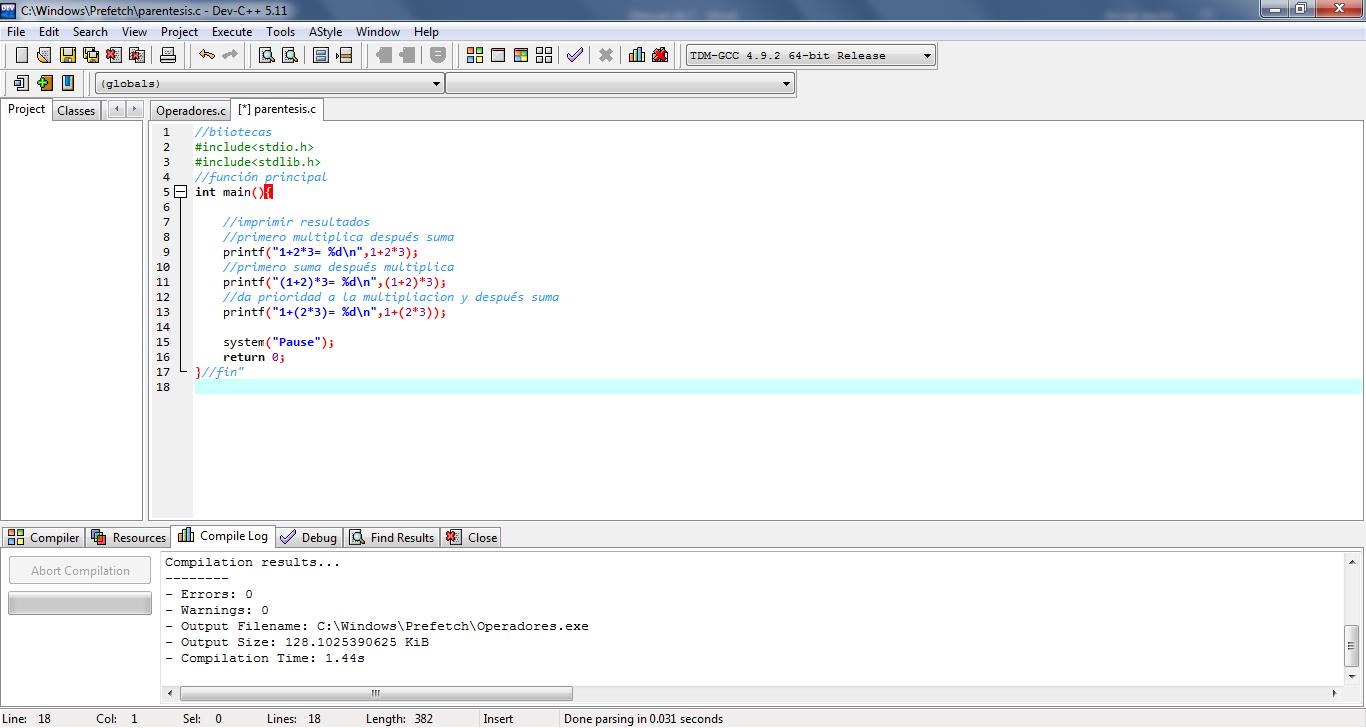
Por ello, al seguir la jerarquía de operaciones y considerando que se utilizan valores solo de tipo entero el resultado es -6



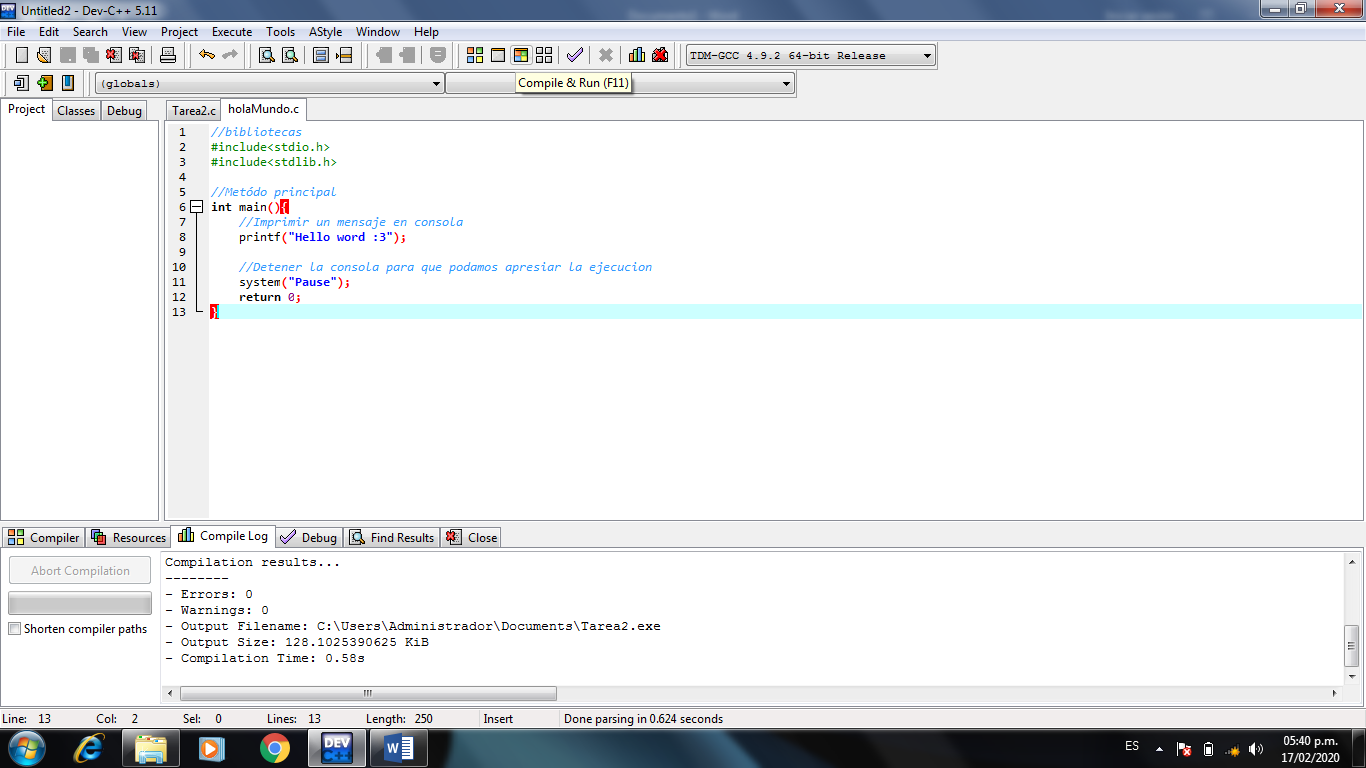
Sin embargo, si existe una forma de dar prioridad a las operaciones, tal es el caso de los paréntesis, como se muestra en este proyecto que nombre parrentesis.c



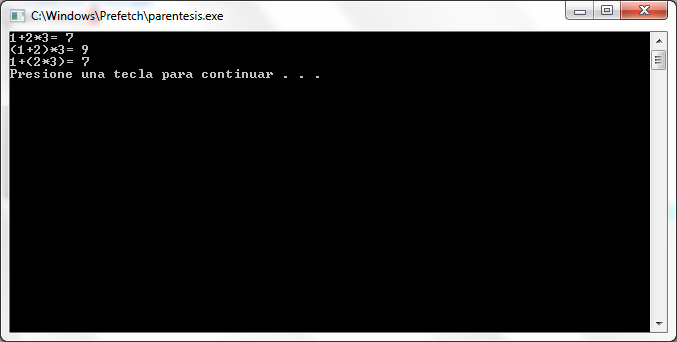
Dentro del código como podemos ver en la primera operación es primero la multiplicación y después la suma, puesto que no se le da prioridad a ninguna de la operación, simplemente siguen el orden, mientras que en la segunda se le da la prioridad a la suma por medio de los paréntesis, Entonces primero se suma el 1+2 y después se multiplica por 3, y finalmente en la última se da prioridad a la multiplicación y después se suma.



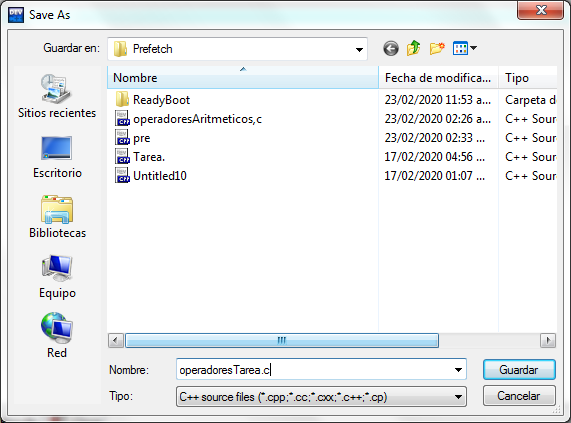
Finalmente ejecutamos el programa por medio de la barra de herramientas



Y este es el resultado de las operaciones



Finalmente, en este tema realizamos un ejercicio el cual nombre opedoresTarea.c

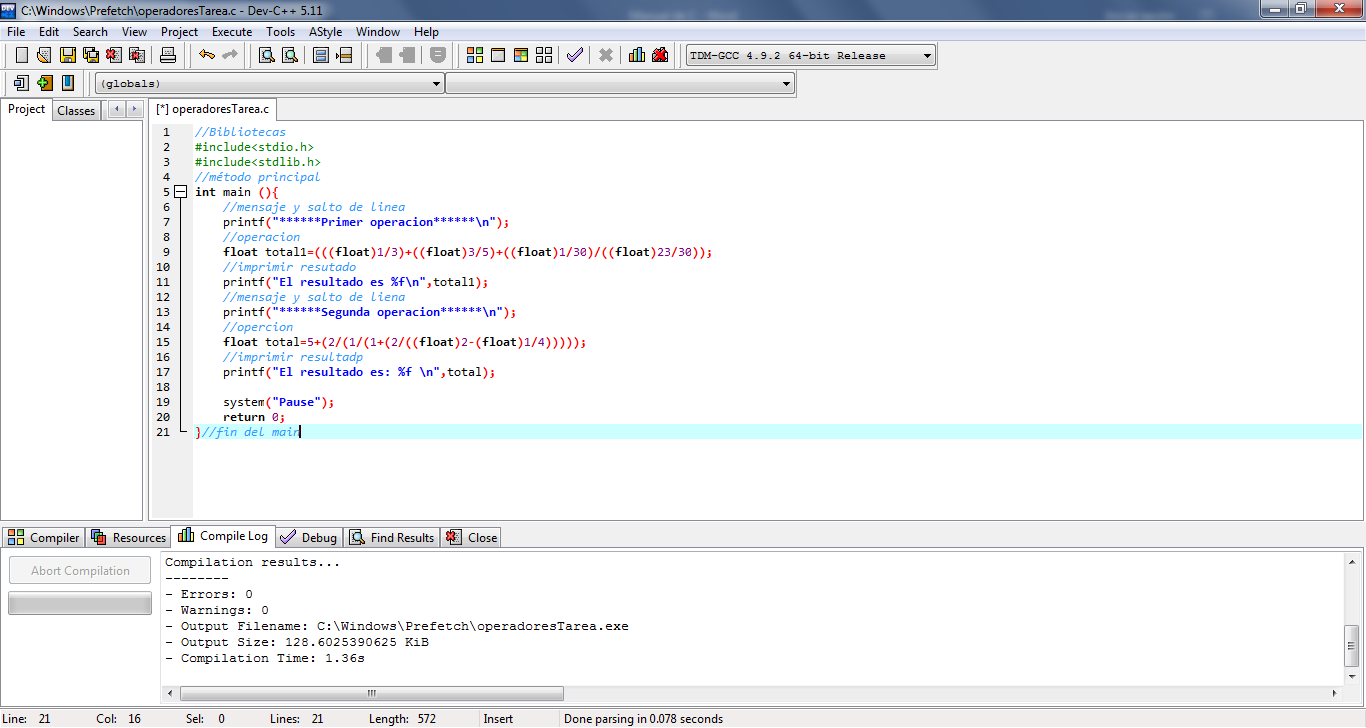


Desarrollo mi siguiente código en el cual utilice variables de tipo float, puesto que trabajare con valores que tienen punto decimal

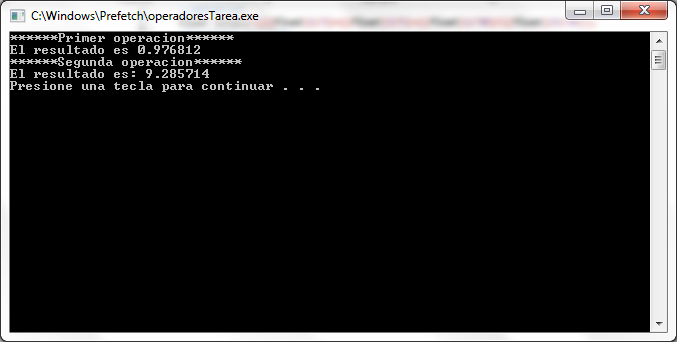
* Hacer un programa en lenguaje c que haga los siguientes cálculos, demuestre los resultados en pantalla.

a) b)

como se puede observar en la primera operación divide 1 entre tres después 1 entre 5 y entre 30, después divide 23 entre 30 y finalmente divide el resultado de la suma de las tres primeras operaciones entre el resultado de la división de 23 ente 30.

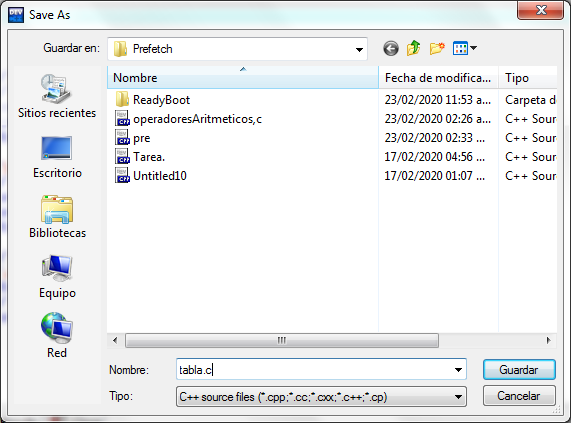


Finalmente, al ejecutar el programa el resultado es el siguiente

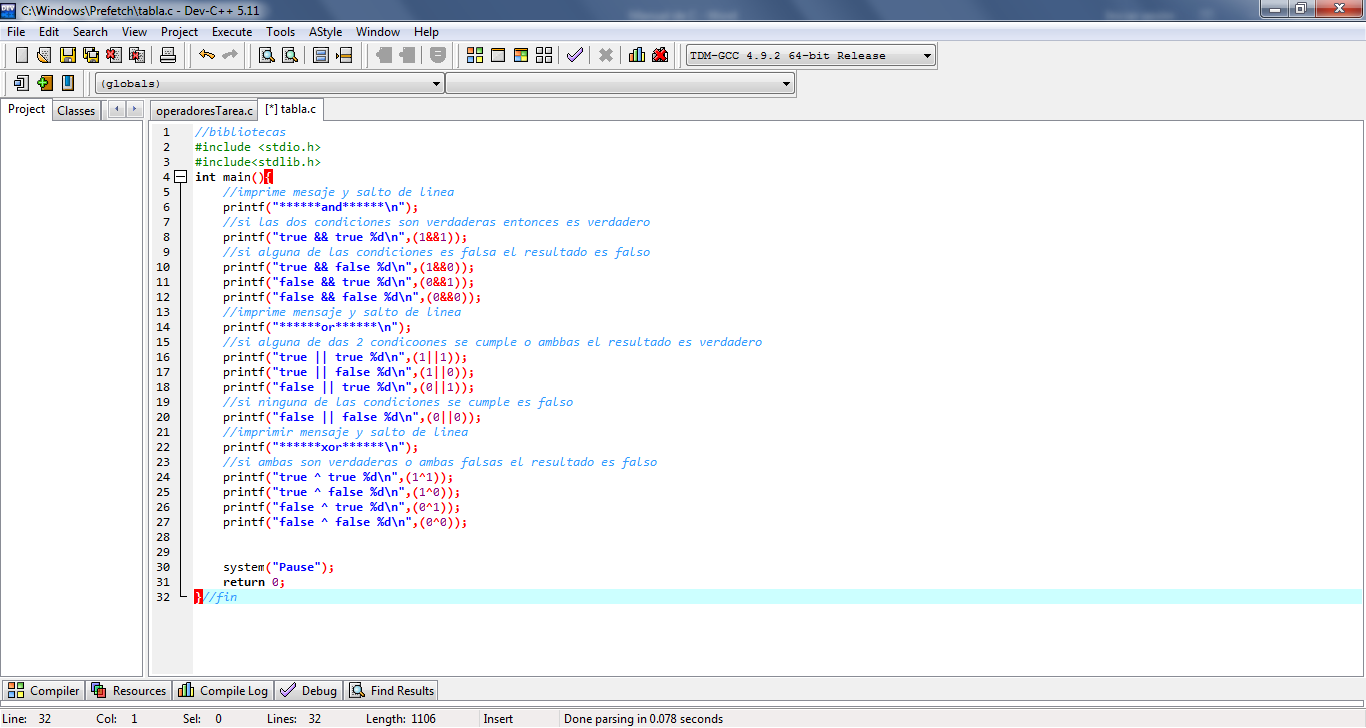


## Lamina 6 Operadores lógicos y relacionales

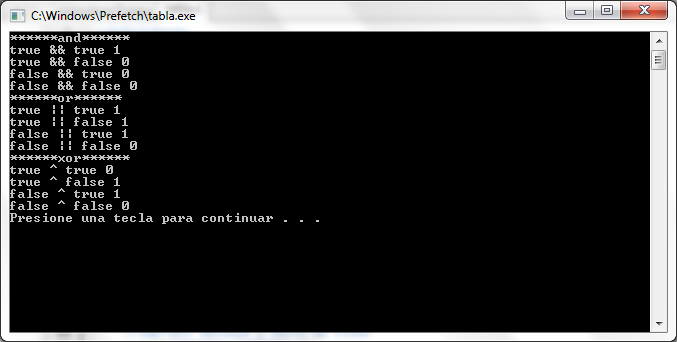
Abrimos un nuevo archivo, el cual nombre tabla.c, ya que mostrara las tablas de verdad para saber si es falso o verdadero, tomando en cuanta el 1 y 0.



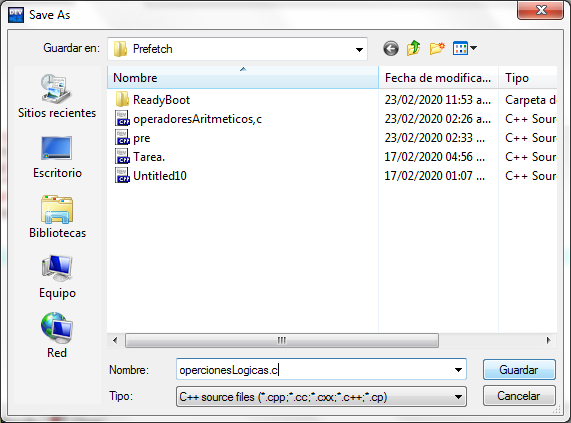
Programamos las siguientes líneas, con la finalidad de mostrar la función de los operadores relacionales como loe el not, xor, and y or



El resultado de la tabla de operadores lógicos es el siguiente, al ejecutarlo en consola se muestra



Posteriormente desarrollaremos un programa, llamado



Mientras que dentro del proyecto ´programamos las operaciones que muestran cómo funcionan dichos operadores evaluando valores en tal caso el 1 es verdadero y 0 falso

En este caso en el primero compara p y r son verdaderos el resultado es **verdadero**

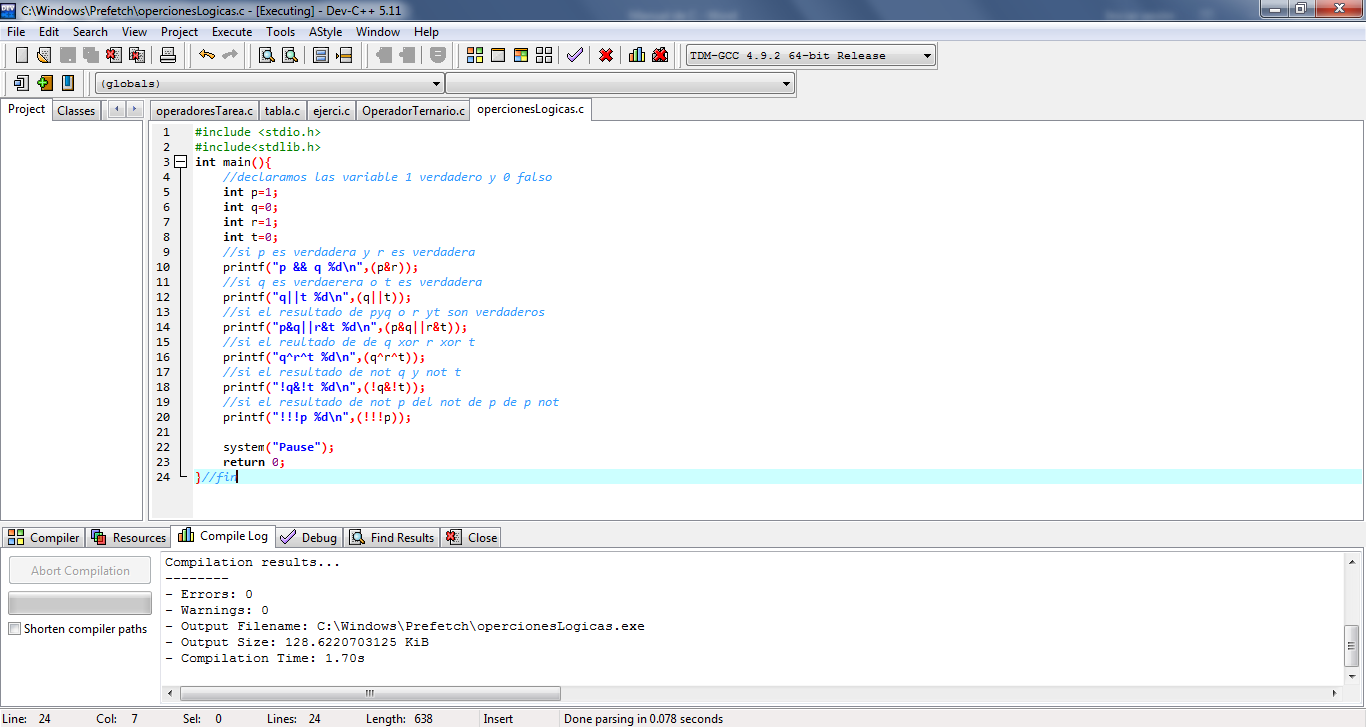
En la segunda línea como ni q no t son verdaderos el resultado es **falso**

En el tercero p es verdadero, pero q no es falso, r es verdadero, pero t no entonces también es falso y como ninguno de los dos resultados es verdadero entonces el resultado final es **falso**

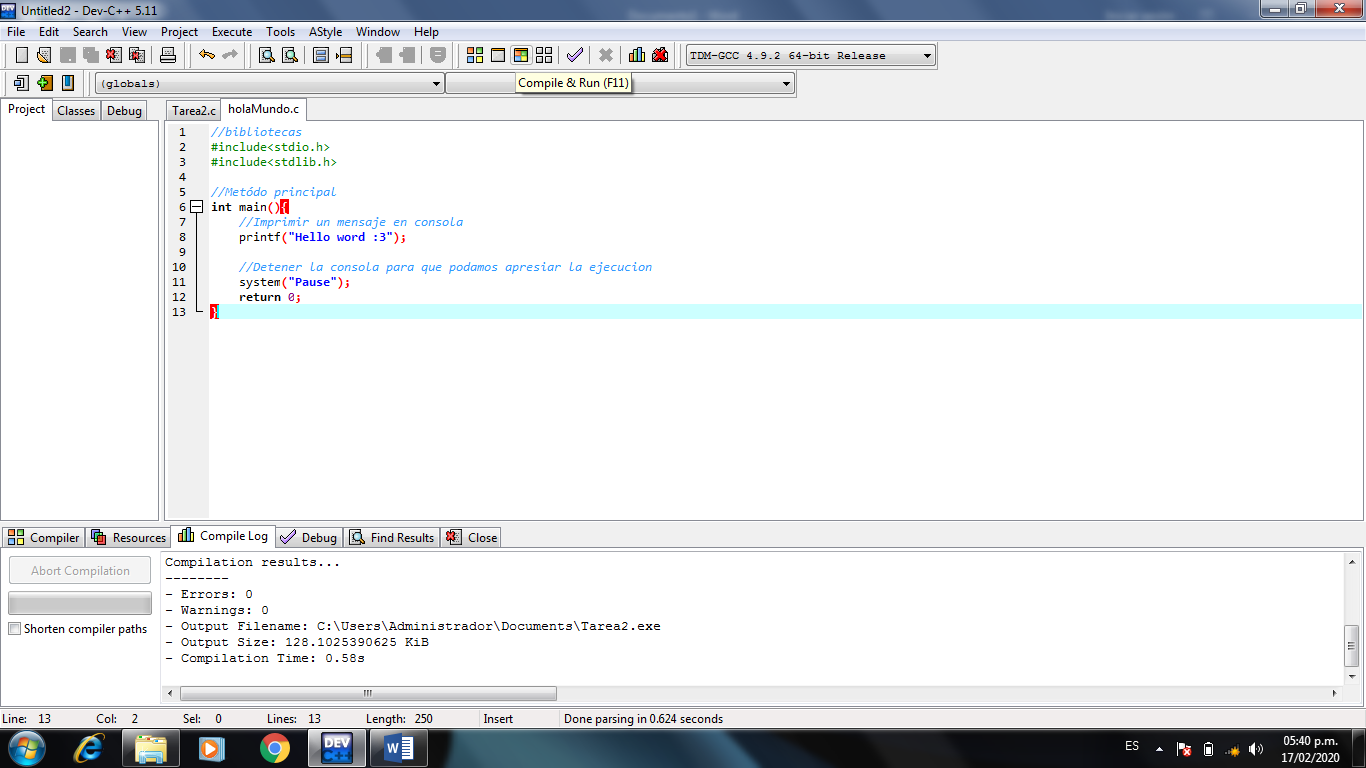
En el cuarto caso como q es falso y r verdadero el resultado con el xor es verdadero y comparado con el xor con t que es falso el resultado final es **verdadero**

Mientras que en el quinto el contrario que es el not de q que en este caso es verdadero y el not de t que es verdadero y en base al and como ambas son verdaderas el resultado es **verdadero**

Y finalmente el contrario de p es falso y el contrario de ese falso verdadero y el contrario falso por lo tanto el resultado es **falso**

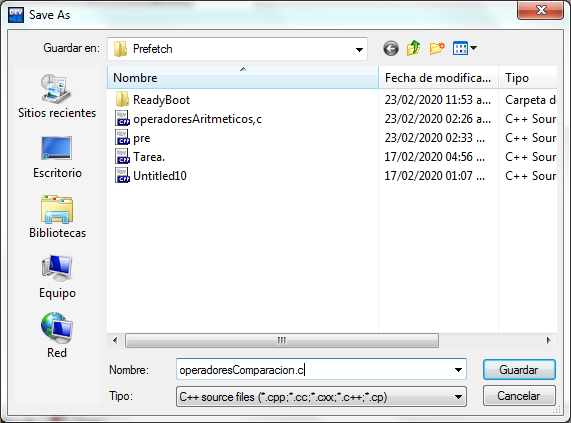


Ejecutamos por medio del icono y se meuestran los resultados en consola





A continuación, trabajaremos con los operadores de comparación, por lo tanto, este será el nombre de nuestro proyecto.



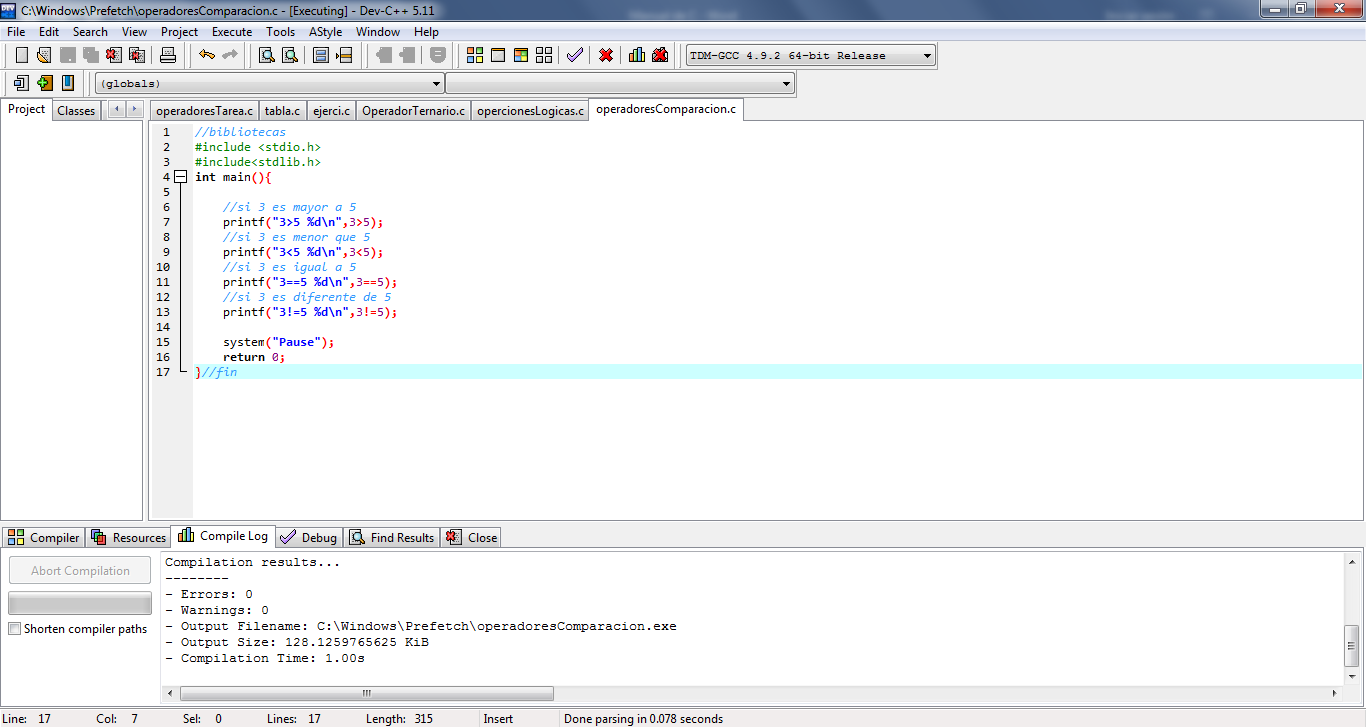
Estos también evalúan valores en su caso trabajan con más valores numéricos no solo 1 y 0, pues en ellos se toman los valores de números comparando si uno es mayor que otro, iguales, diferentes menores menor igual o mayor igual y nos dice si estos se cumplen o no por medio de un falso o verdadero, tal es el caso del siguiente código.

En la primera línea comparamos si 3 es mayor a 5 y el resultado es falso

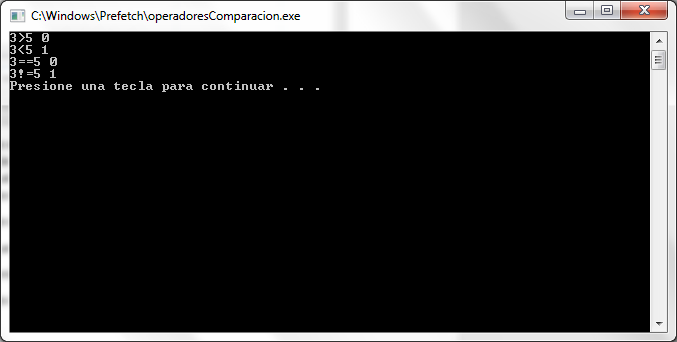
En la segunda si 3 es menor que 5 y el resultado es verdadero

En la cuarta comparamos so ambos valores son iguales y el resultado es falso

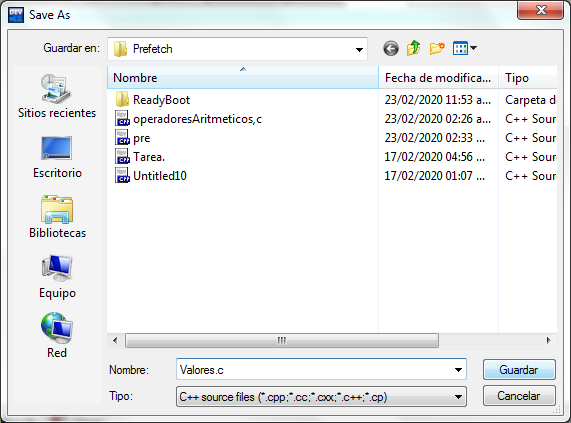
Y finalmente se compara si 3 es distinto a 5 y el resultado es verdadero



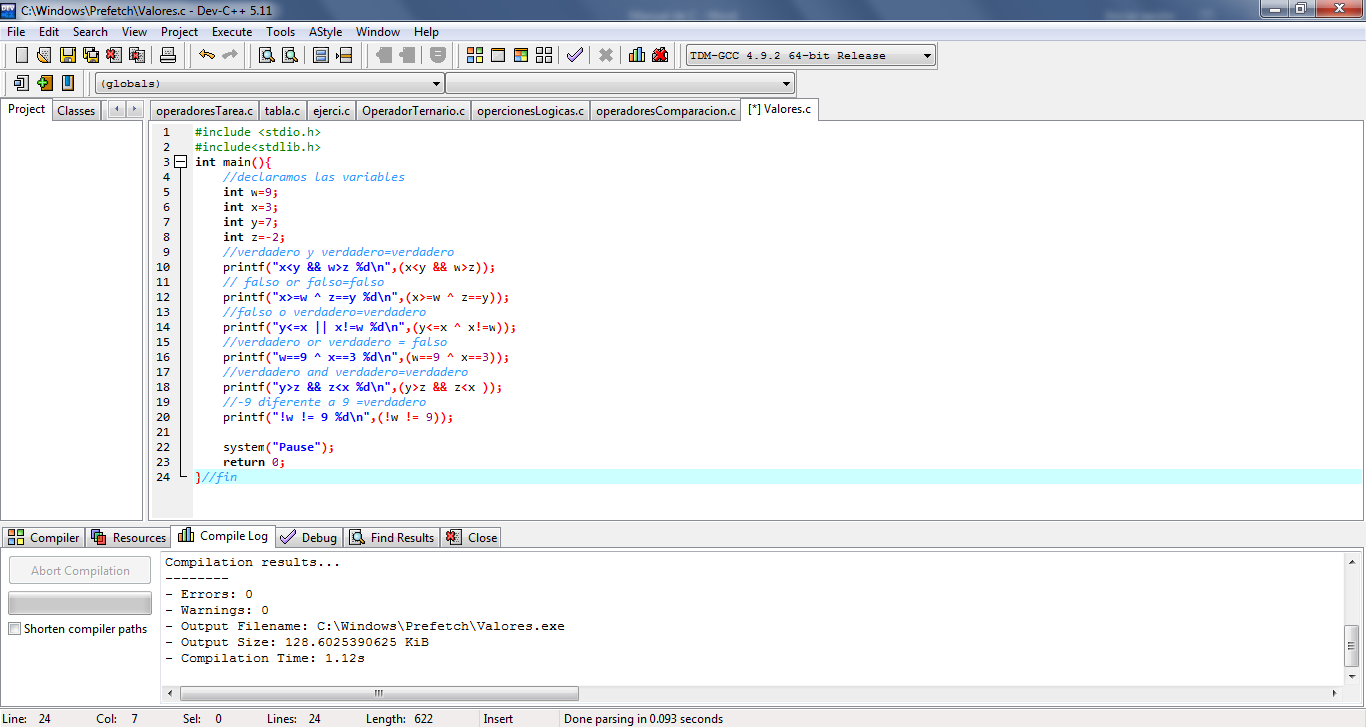
Por lo tanto, al ejecutar y compilara el proyecto en consola este es el resultado con o si es falso y 1 si es verdadero



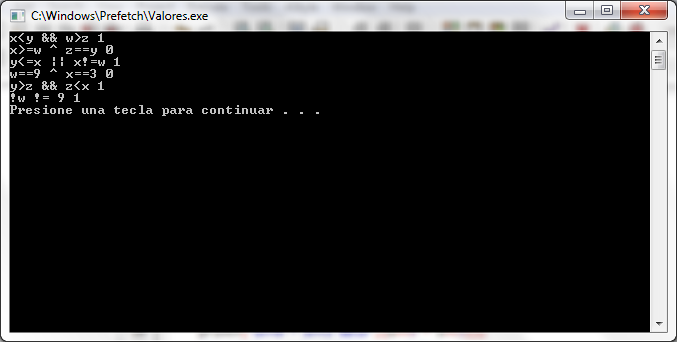
Para finalizar con esta lamina creamos un nuevo programa llamado valores.c



En el área de trabajo realizamos las siguientes operaciones para saber si el resultado es falso o verdadero, acorde a cómo funcionan los operadores lógicos y relacionales según sea el caso como se ha visto hasta ahora

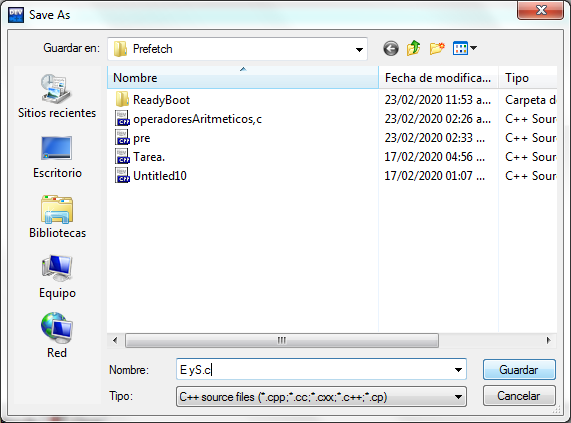


Ejecutamos nuestro programa y este es el resultado en consola.

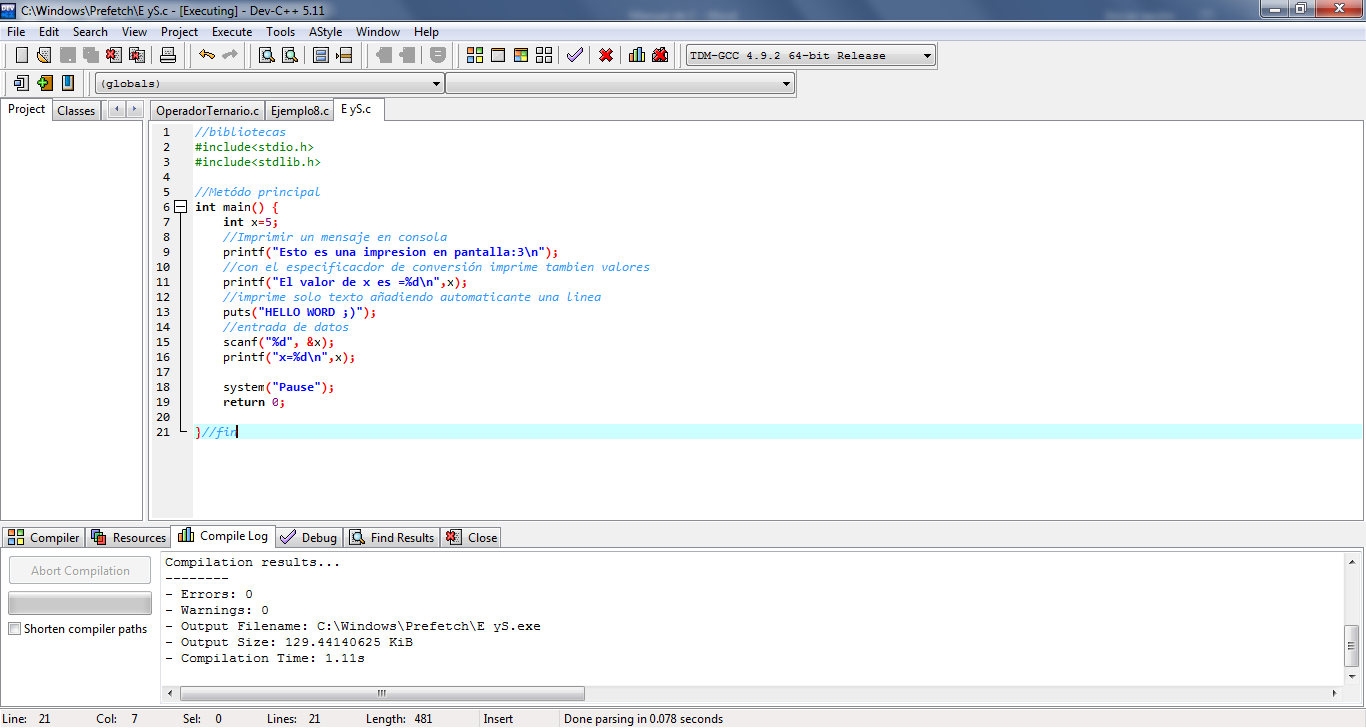


## Lamina 7 Entrada y salida de datos

Creamos un nuevo proyecto llamado EyS . c



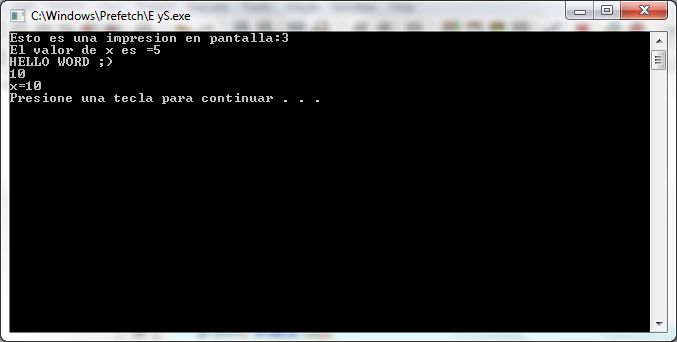
N el escribimos un código para diferenciar la estructura de cuando entra o sale un dato de la siguiente manera



Finalmente, al ejecutarlo en consola aparece esta pantalla imprimiendo las líneas de código de salida de datos



Sin embargo, si ingreso cualquier valor será el que tomará para x y ejecuta la línea 15 y 16 de mi código como se muestra a continuación



1. **Conclusiones:**
2. Por medio de las practicas analizamos la sintaxis para declaración de variables, según sus tipos de datos para optimización de recursos y así ahorrar espacio en memoria, de igual manera aprendí a imprimir mensajes con y sin valores, al igual que la entrada y salida de datos, así realizar operaciones básicas siguiendo la jerarquía de operaciones, al igual que comprobar dichos procesos obteniendo los resultados esperados,