|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre de la práctica** | **Laminas 8-11** | | | **No.** | **2** |
| **Asignatura:** | **Métodos numéricos** | **Carrera:** | **Ing. Sistemas Computacionales** | **Duración de la práctica (Hrs)** |  |

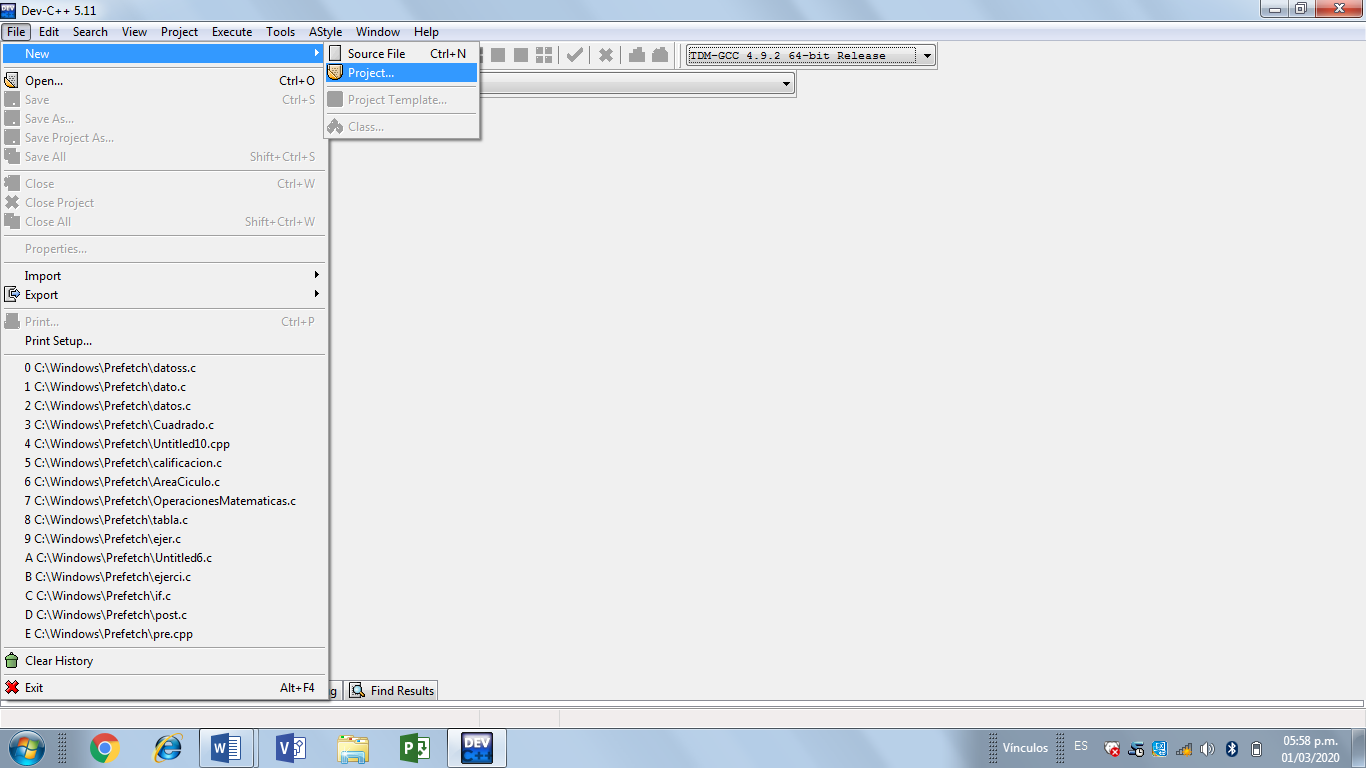
**II. Lugar de realización de la práctica (laboratorio, taller, aula u otro):** Otro

**III. Material empleado:** laptop y Dev-c

**IV. Desarrollo de la práctica:**

# Lamina 8 if

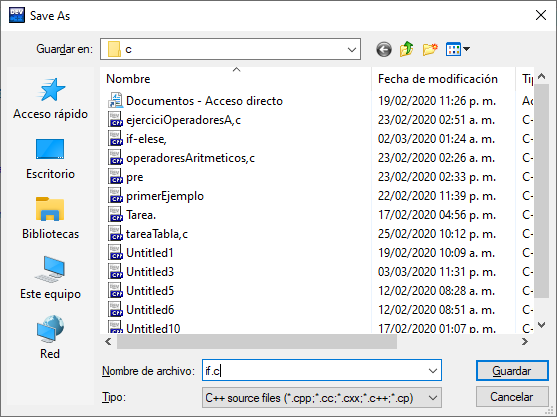
En esta práctica lo primero a realizar es abrir el programa mediante el acceso directo dando clic derecho sobre su icono y abrir, posteriormente en la parte superior en la barra de herramientas damos clic derecho cobre file después new y proyecto para crear de esta forma un nuevo archivo



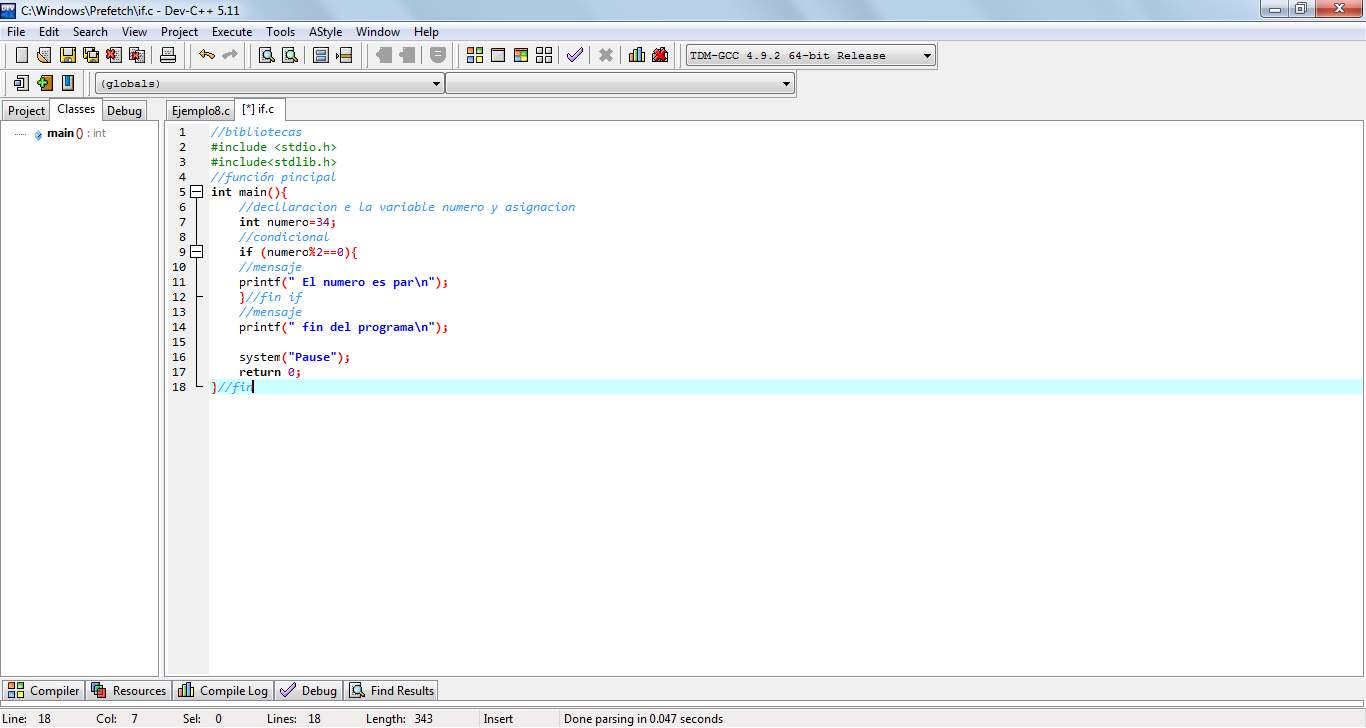
De igual manera guardare este proyecto con un nuevo nombre.



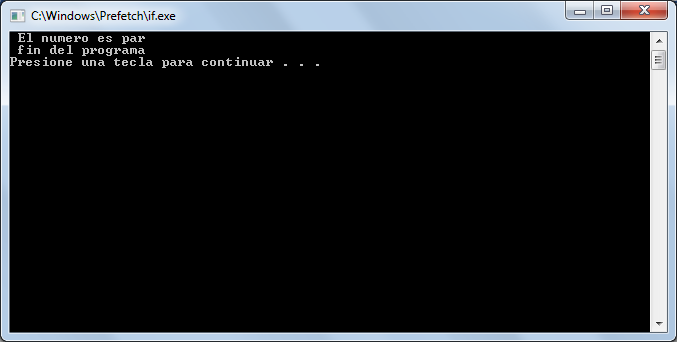
Este proyecto se guarda con la extensión .c



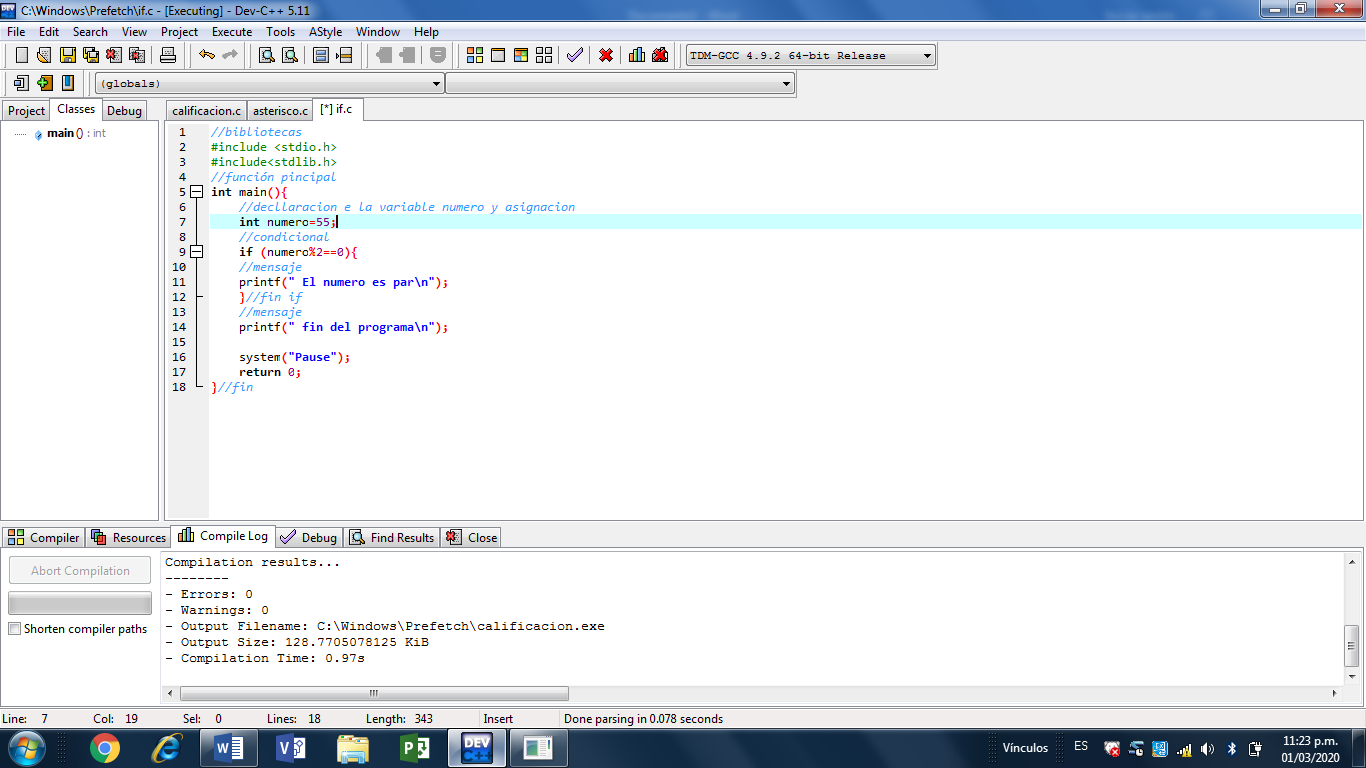
Este programa evalúa si la variable es par o impar en este caso divide entre 2 y si el residuo de este número es igual a 0 entonces envía un mensaje indicando que es par y finaliza.



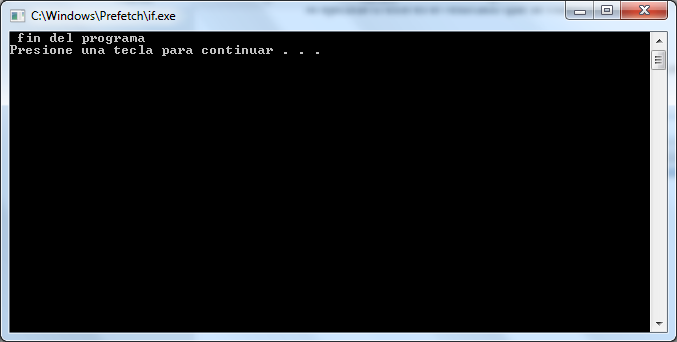
Al compilar el programa se muestra en pantalla



Ahora este mismo programa lo modificaremos asignando a la variable numero el valor de 55, por lo tanto, ya no sería un numero par

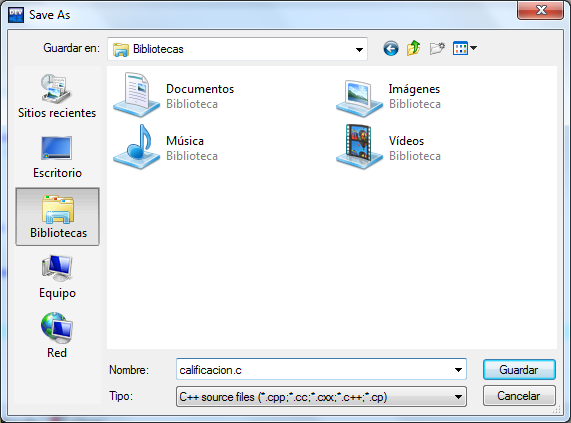


Al ejecutarlo este es el resultado que se muestra en consola, ya que al ser evaluado no cumplió con la condición, y por lo tanto no se ejecutó el mensaje dentro de ella



* Escriba un programa que reciba por teclado la calificación del examen de un alumno e imprima el siguiente texto únicamente si la calificación es aprobatoria
* Mensaje:” Felicidades aprobaste el curso”
* La calificación mínima aprobatoria es de 8

Cree un nuevo programa al cual llame calificación.c

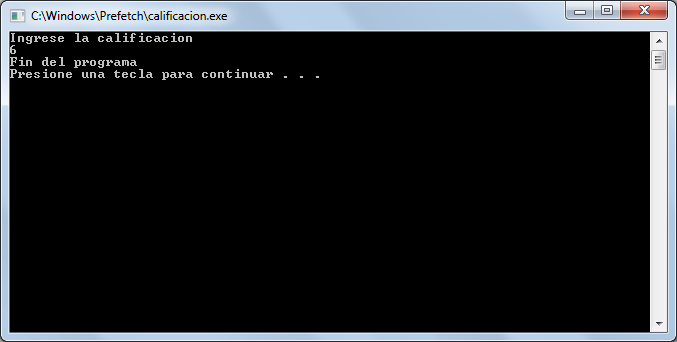


Una vez dentro del área de trabajo desarrolle el siguiente código, el cual consiste en evaluar si una calificación asignada es aprobatoria o no.

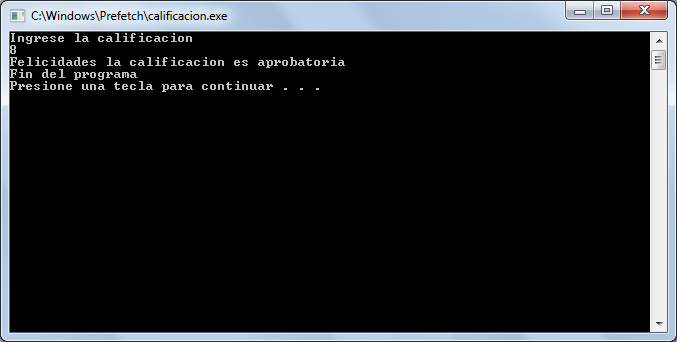
Primero declara la variable donde se almacenará el valor ingresado por el usuario, posteriormente solicita por medio del mensaje que se ingrese la calificación, este valor será asignado a la variable cal, posteriormente es evaluado dicho valor donde si es mayor o igual a 8 se ejecuta el mensaje del if "Felicidades la calificación es aprobatoria", en caso contrario solo ejecuta el mensaje fuera del condicional if “Fin del programa”



Al ejecutarlo ingresare una calificación de 6 para ver la ejecución al no cumplirse la condición

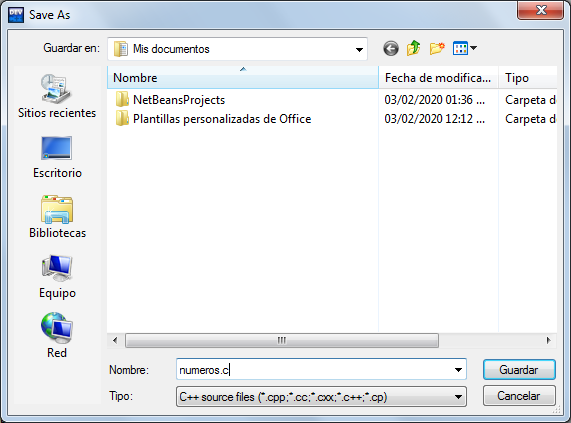


Mientras que al ingresar 8 la condición se cumple

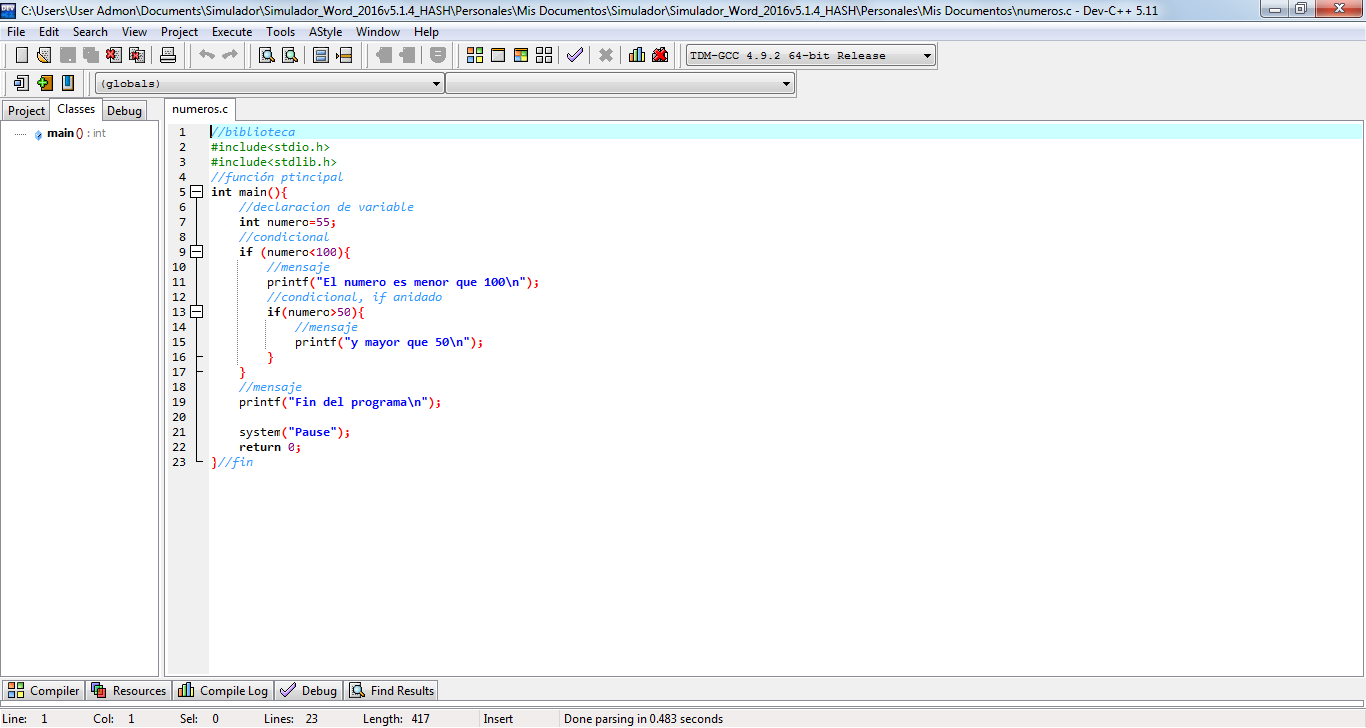


Mediante el siguiente código se mostrarán la función y sintaxis de un if anidado, el cual consiste en avaluar 2 o más veces un valor.

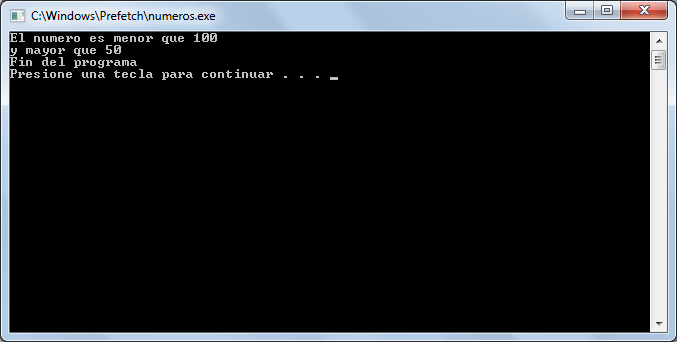
A este nuevo programa lo nombre números.c



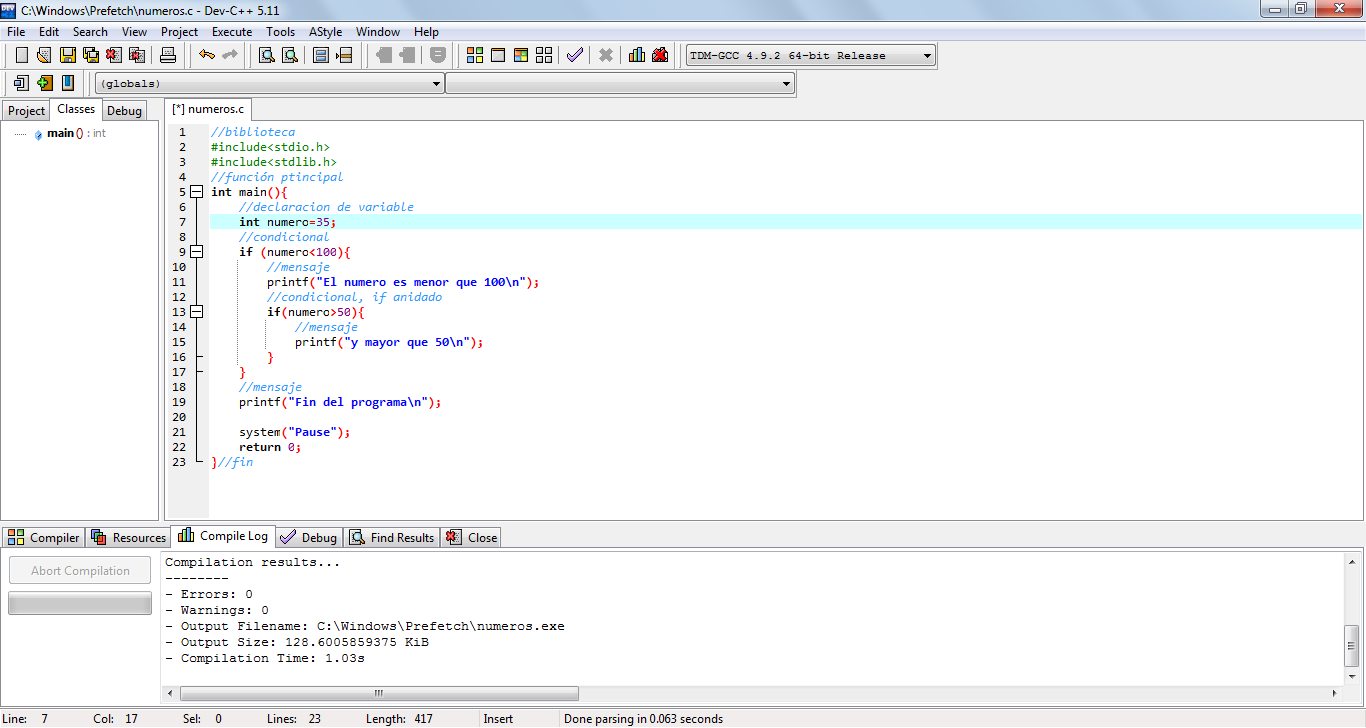
Una vez dentro del área de trabajo desarrolle el siguiente código el cual consiste en evaluar la variable numero la cual tiene el valor de 55, por lo tanto, en la primera condicional evalúa si 55 es menor a 100, lo cual es verdadero, por lo tanto, imprime el primer mensaje, posteriormente evalúa a la variable si 55 es mayor a 50, lo cual es verdadero, por ello se imprime el segundo mensaje y finalmente se termina de ejecutar el programa



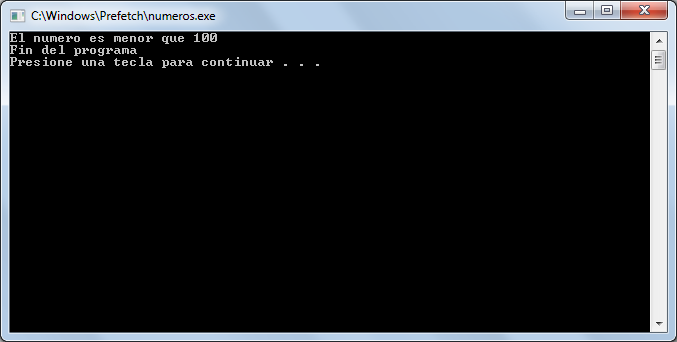
Al ejecutar el programa se muestran los mensajes, indicando que el número cumplió con ambas condiciones



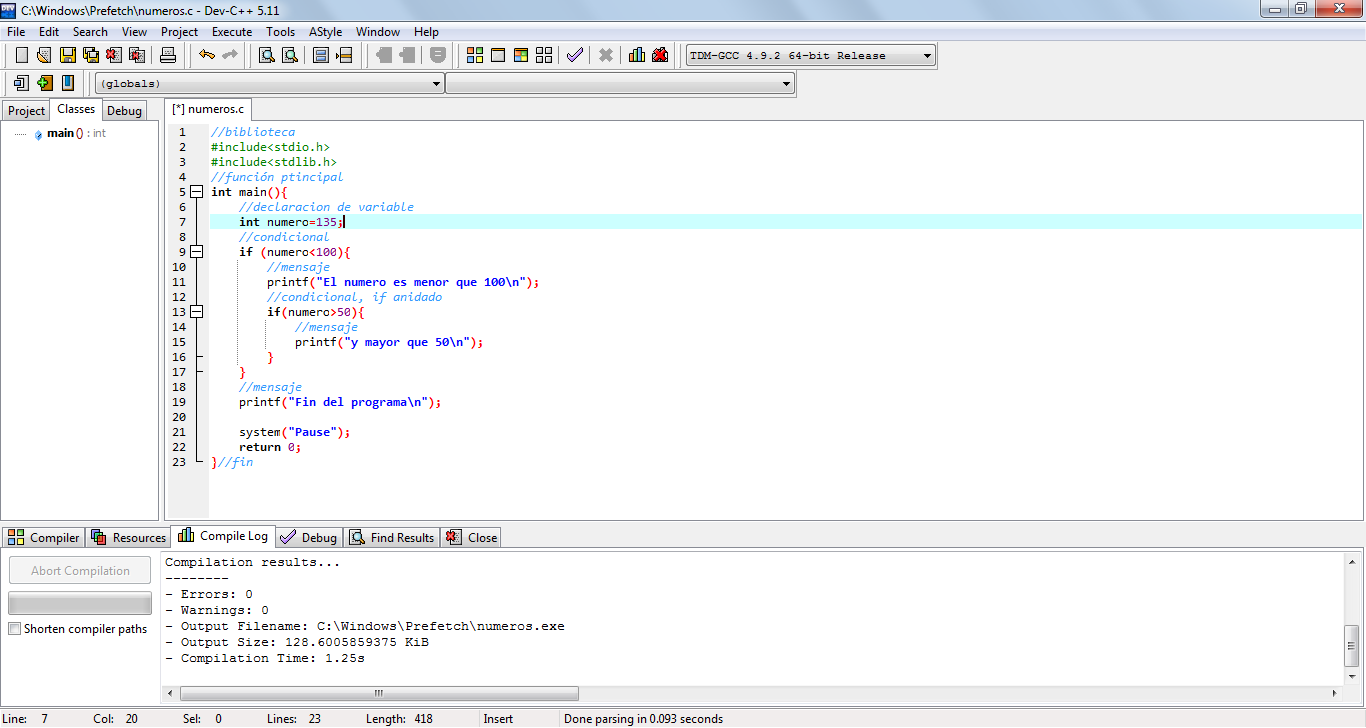
Ahora modificare el valor asignado a la variable número, por 35como se muestra



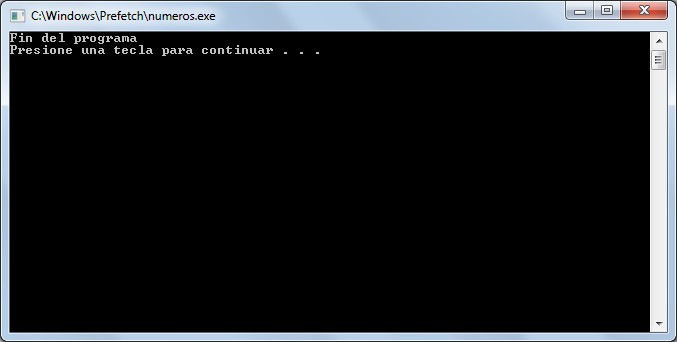
Al ejecutarlo se muestra solo uno de los mensajes de los if lo cual indica que solo se cumplió una condición ya que el numero si es menor a 50 pero no es mayor a 50.



Posteriormente este valor de la variable lo modificare nuevamente, ahora le asigne el valor de 135, como se muestra.



Sin embargo, a pesar de cumplir el segundo condicional, no se muestra el mensaje, ni mucho menos es evaluado el numero por este segundo if, puesto que, si no se cumple con el primer condicional, no se ejecutan las líneas de código dentro del, por lo tanto, solo se muestra el mensaje que esta fuera de los 2 condicionales.

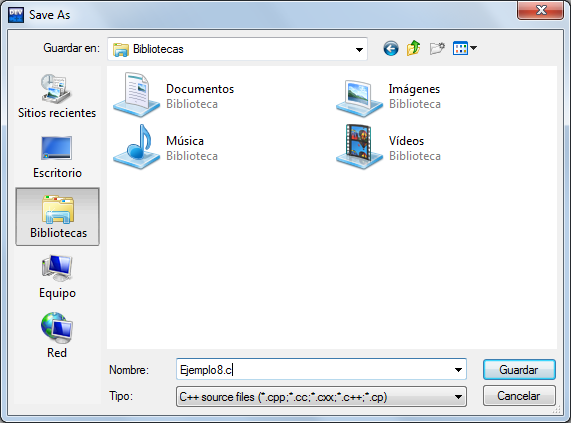


Creo un nuevo programa y los guardo como ejemplo8 como anteriormente se realizó con la extensión .c

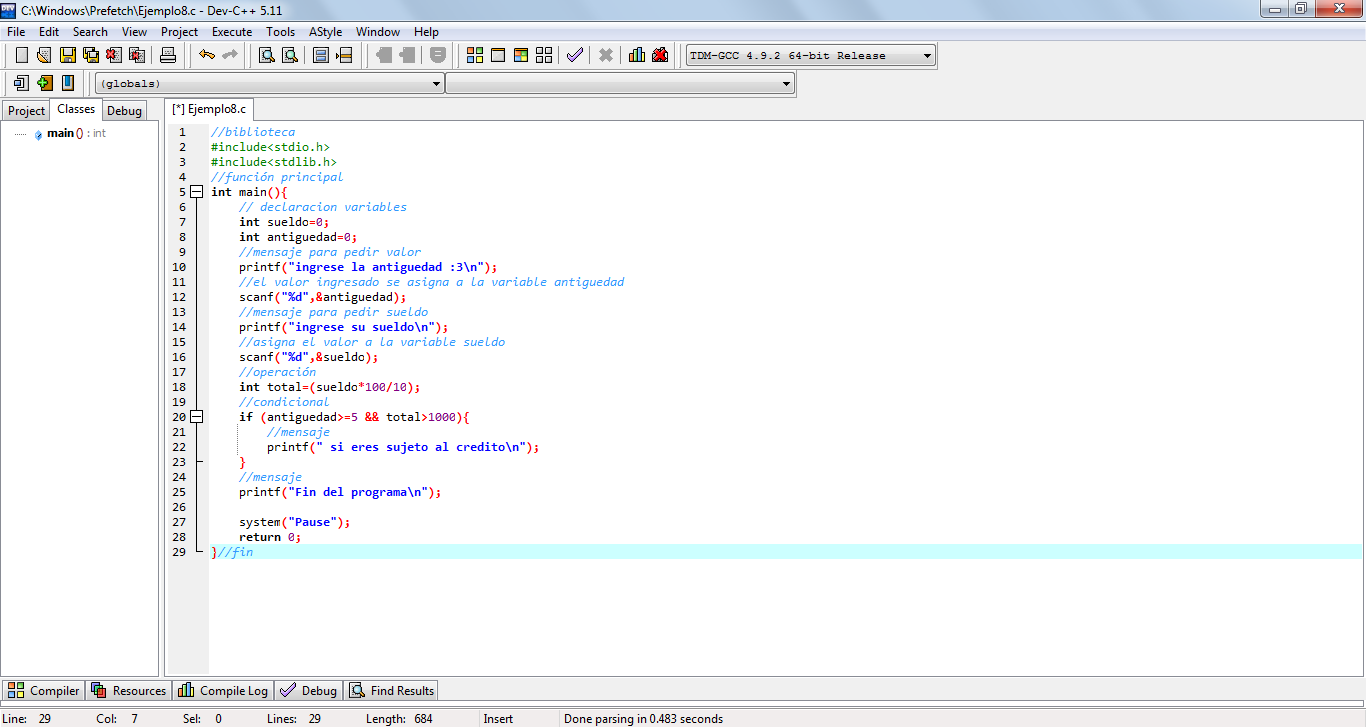
Realizar un programa, que te diga si la persona es sujeta a un crédito hipotecario.

Los condicionales que debe cumplir son:

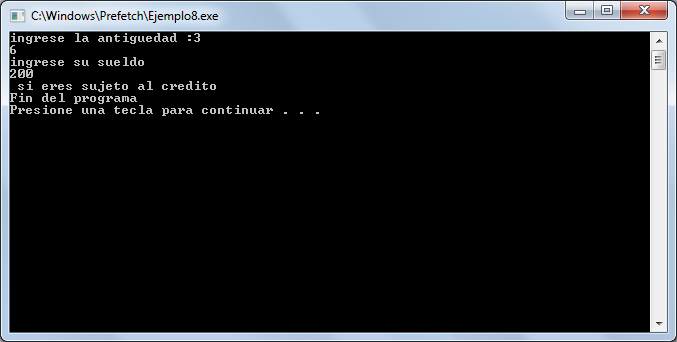
* Tener 5 años de antigüedad en el trabajo.
* El 10% de su sueldo mensual debe ser mayor a 1,000 pesos



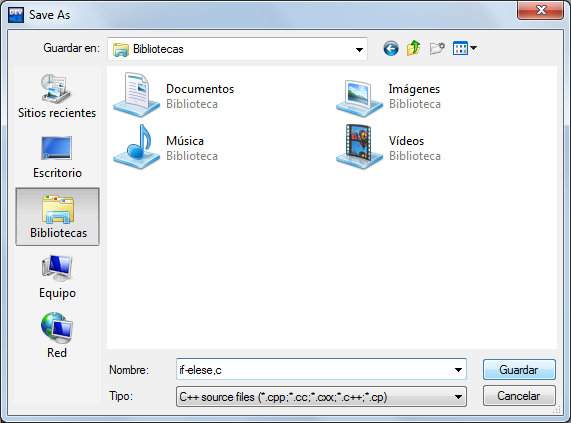
El programa solicita que se asignen el valor de antigüedad, así como el sueldo por hora, lo primero que realiza es obtener el equivalente al 10% de su sueldo, mediante una operación y es signado a la variable total, después evalúa, si la variable antigüedad posee un valor mayor a 5 y si la variable total, posee un valor mayor a 1000, entonces ejecuta el mensaje dentro de la condicional, pero si no entonces solo imprime el mensaje fuera de la condición.



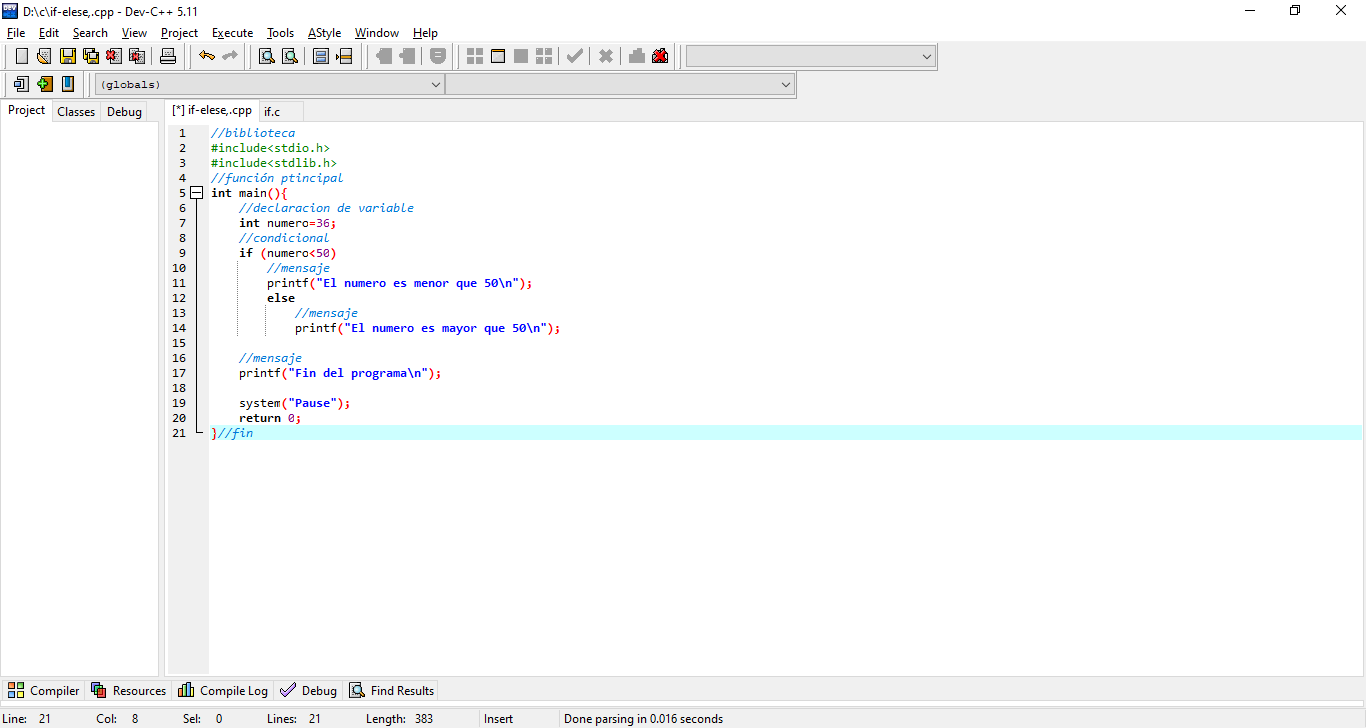
Al ejecutar el programa ingrese los siguientes valores, para probar la ejecución del if



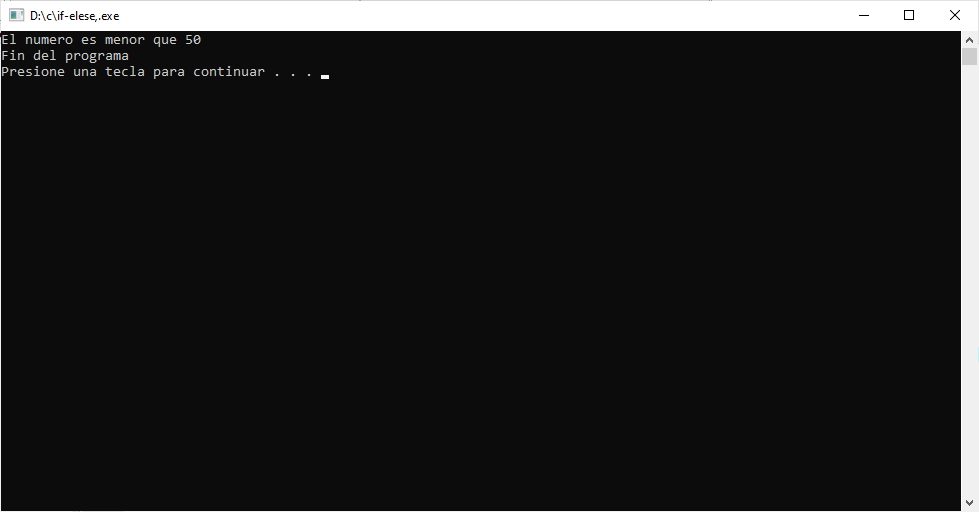
A continuación, creo un nuevo proyecto, el cual nombre if-else.c como se puede observar.



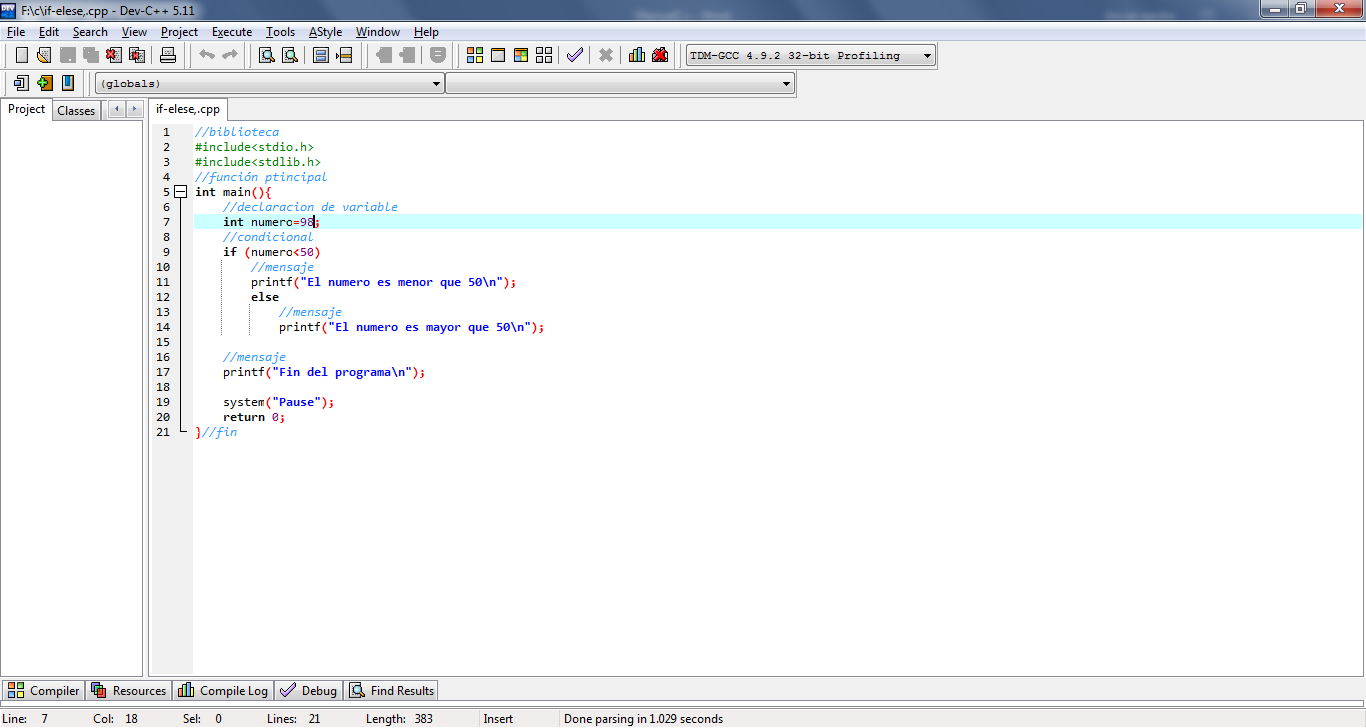
Dentro del espacio de trabajo programe las siguientes instrucciones, en las cuales asigna el valor de 36 a mi variable y mediante la condición if será evaluada si el numero es menor a 50 se ejecutará la instrucción dentro del if que es el mensaje indicando que el numero es menor a 50, en caso contrario se ejecuta la instrucción del else en el cual tenemos el mensaje "El numero es mayor que 50”



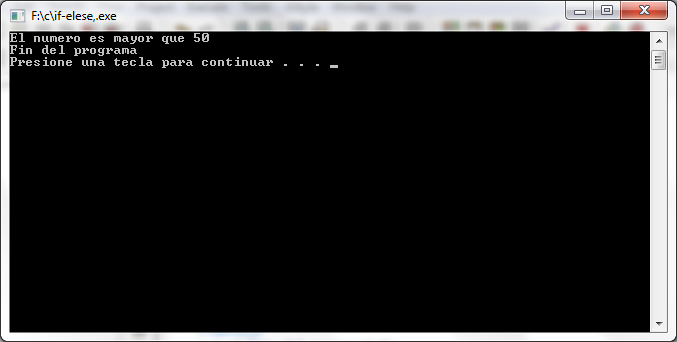
Una vez analizado el procedimiento ejecuto mi programa en el cual el resultado debido a que el valor a evaluar es 36 se debe ejecutar el mensaje dentro del if ya que 36 es menor que 50.



Posteriormente este mismo programa lo reutilice modificando el código para que el valor asignado a la variable numero ahora sea de 98



Por lo tanto, al ser evaluada la variable con este nuevo valor no entra dentro del if y por ello ejecuta el else, como resultado se envía el mensaje en consola.



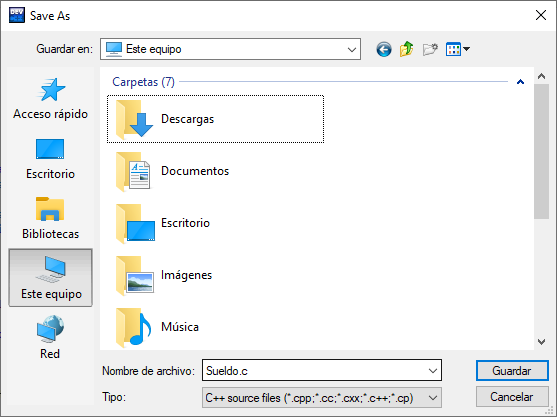
Calcular el salario semanal de un empleado, sabiendo que éste se calcula en base a las horas semanales trabajadas y de acuerdo a un precio especificado por hora. Si se pasa del as 40 horas semanales, las horas extras se pagarán a razón de 1.5veceslahoraordinaria.

ENTRADAS: Horas semanales trabajadas (HST) Precio por hora (PH)

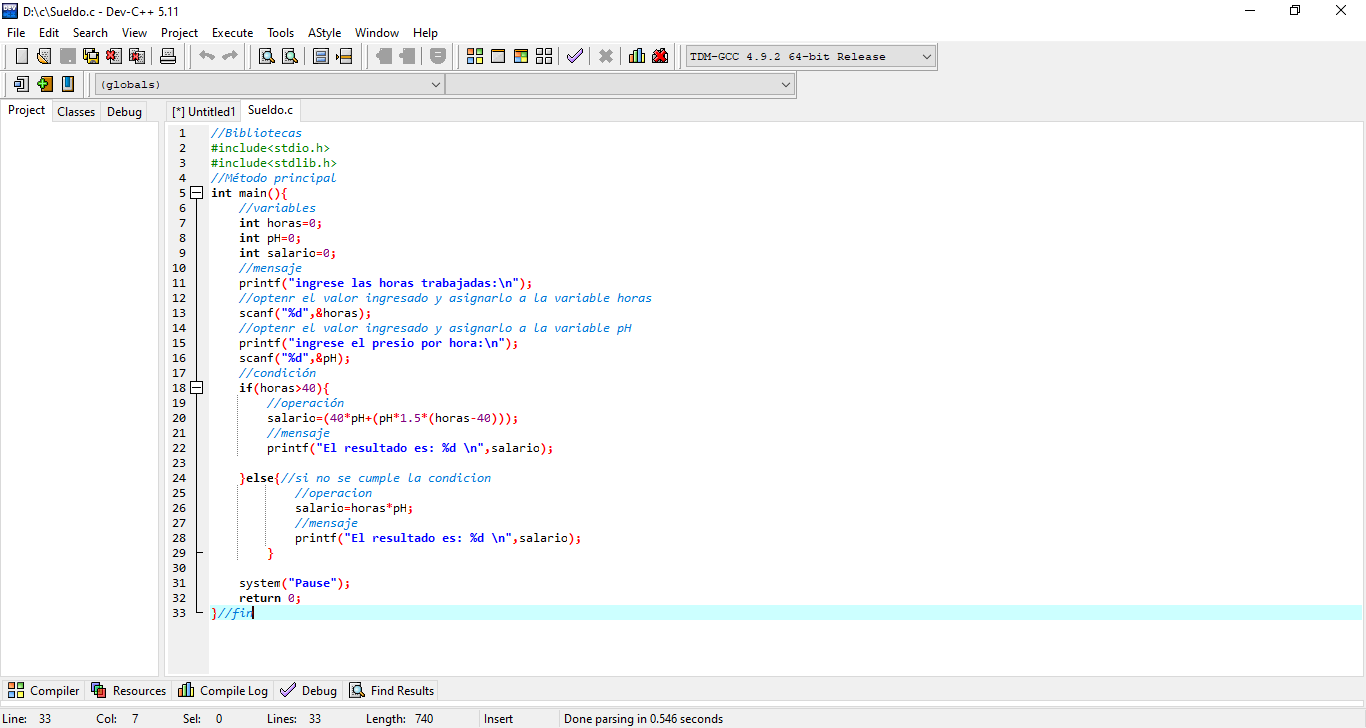
PROCESO: Si las horas semanales trabajadas <= 40 Calcular salario semanal = (HST \* 40)

Si las horas semanales trabajadas > 40 Calcular salario semanal = (40 \* PH + (PH \* 1.5 \*(HST-40)))

Creo un nuevo proyecto con el nombre de sueldo y la extensión. C en el cual desarrollaremos nuestro proyecto.



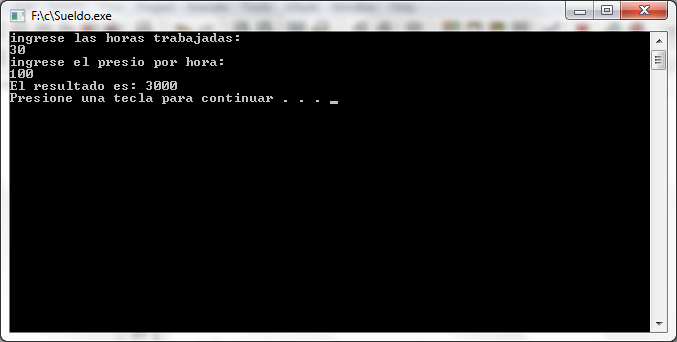
Primero solicitamos que el usuario ingrese las horas trabajadas y el precio por hora, una vez obtenidos los valores evaluamos si las horas son mayores a 40 de ser así se ejecutan las líneas dentro del primer if, pero si no entonces se ejecutan las líneas del else



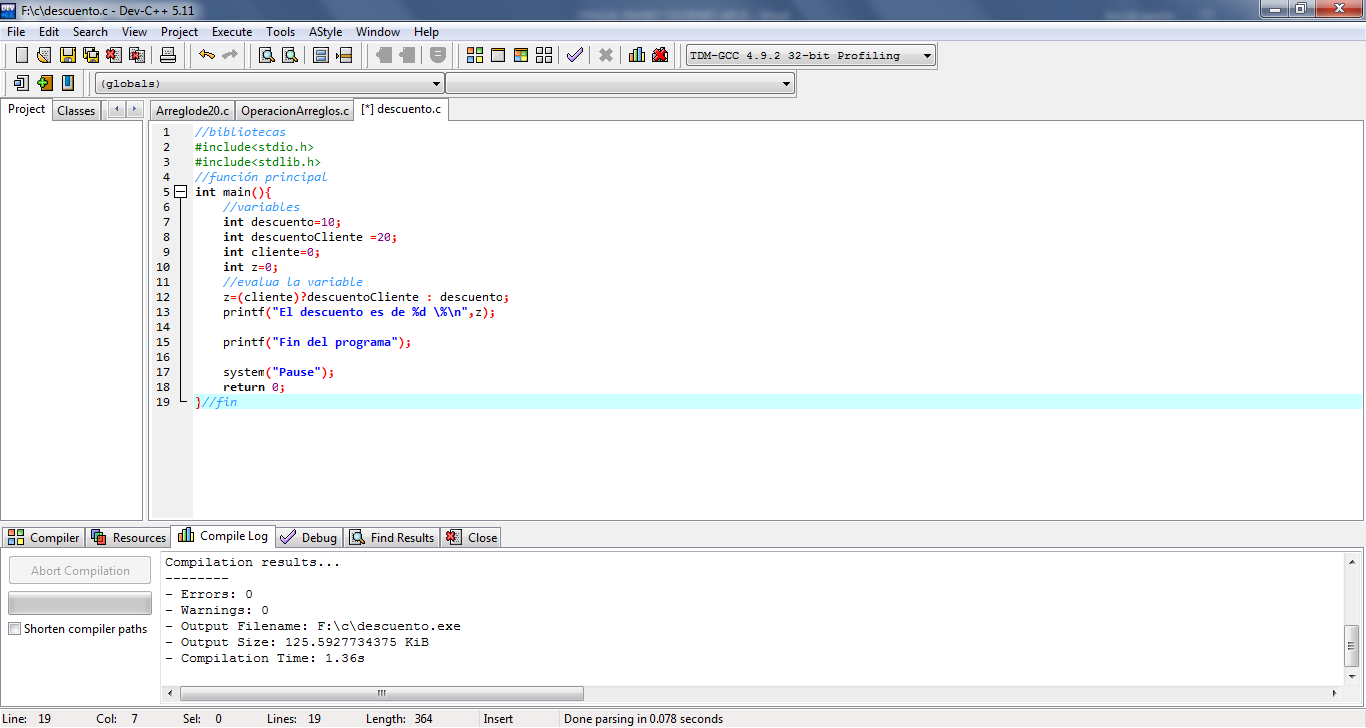
Al ejecutarlo primero ingrese un valor mayor a 40 para verificar la ejecución del if como se muestra en consola



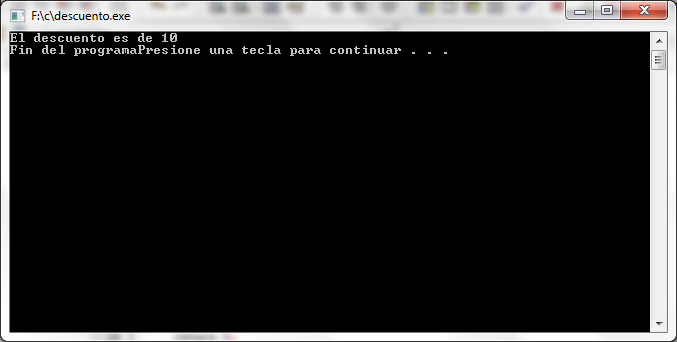
Mientras que al volver a ejecutarlo ingrese un valor para horas menor a 50, por lo tanto, se ejecutan las instrucciones del else



Finalmente se muestra el siguiente ejemplo en el cual evaluamos según el valor de 1 y 0 asignado a el valor de cliente

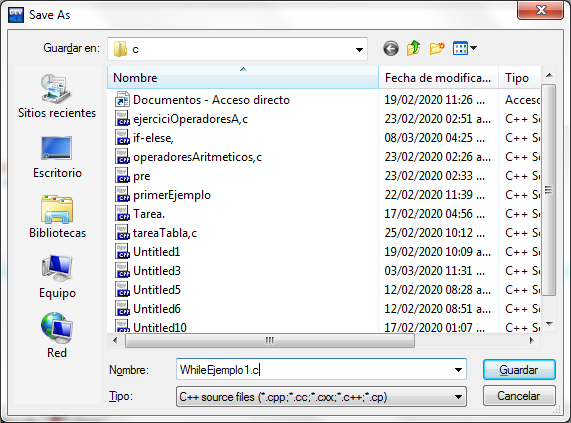


En este caso como cliente vale 0 la variable z toma el valor de descuento y se mostrara el siguiente resultado en pantalla



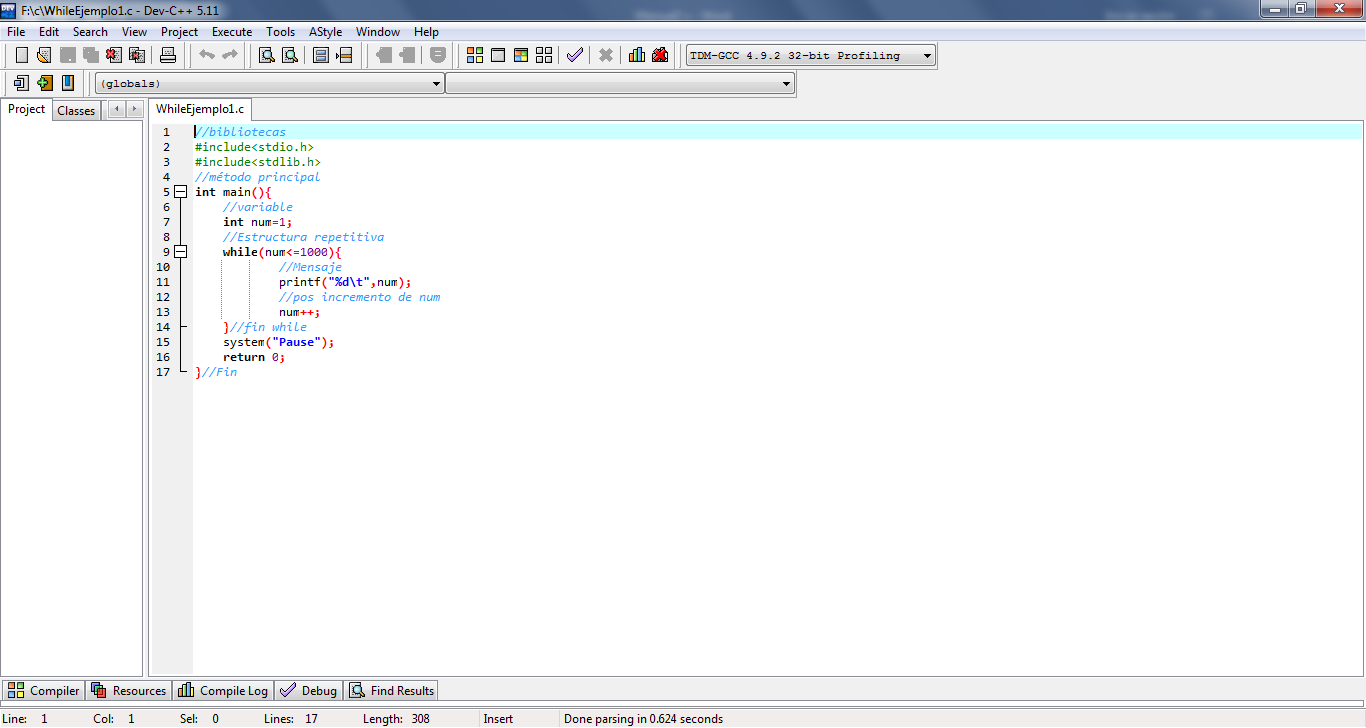
# Lamina 9 While

Creamos un nuevo proyecto como se ha realizado anteriormente, el cual llamaremos WhileEjemplo y del mismo modo lleva la extensión .c

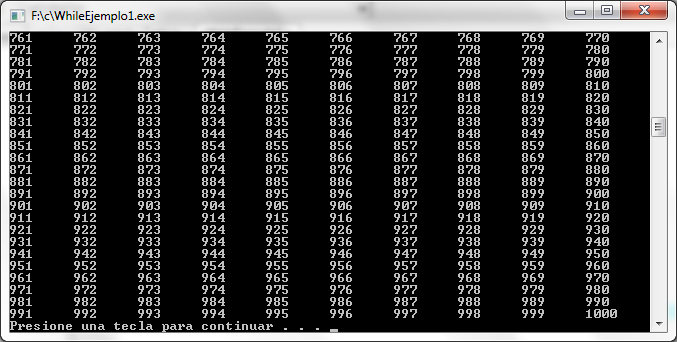


Posteriormente coloque una sentencia de instrucción en la cual evaluamos nuestra variable numero inicializada en uno, el ciclo while funciona de la siguiente manera:

La variable num es evaluada y las instrucciones dentro del ciclo while si es menor 1000 y en caso contrario se termina



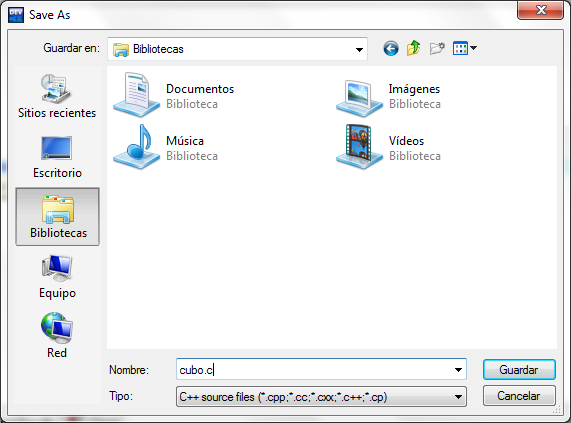
Al ejecutar el programa se muestra de la siguiente manera



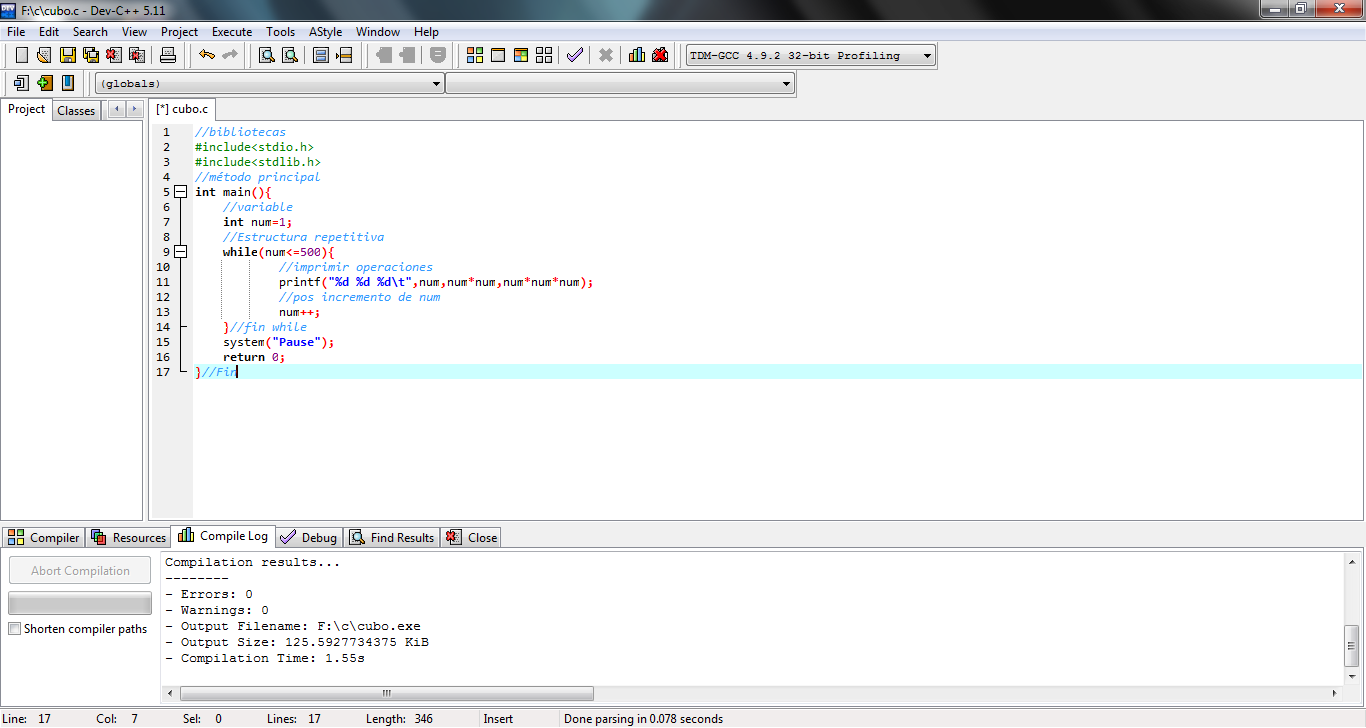
Imprimir los cuadrados y los cubos de los primeros quinientos números naturales

* El cuadrado de un número se define como: x²=x\*x
* El cubo de un número se define como: x³=x\*x\*x

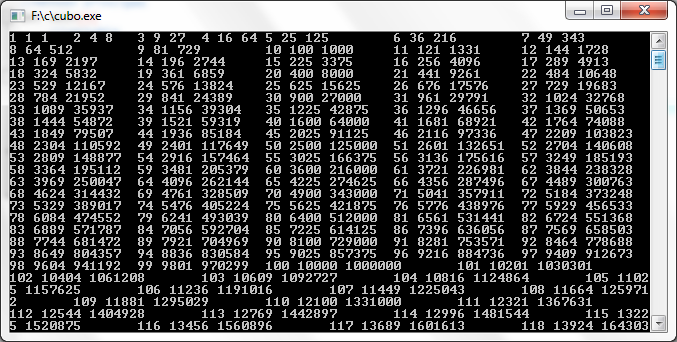
Primero creo y guardo mi proyecto como cubo.c



Para este ejercicio cada vez que la variable num tenga un valor menor o igual a 500 se imprimirán el valor de num, así como la multiplicación de su valor actual por sí mismo y el valor por sí mismo y nuevamente por num para obtener el cubo de este valor y finamente se incrementa la variable num como se muestra



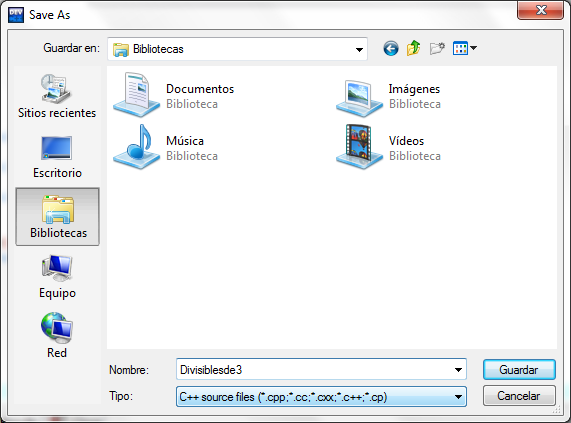
Este es el resultado al compilara el programa y detener la consola



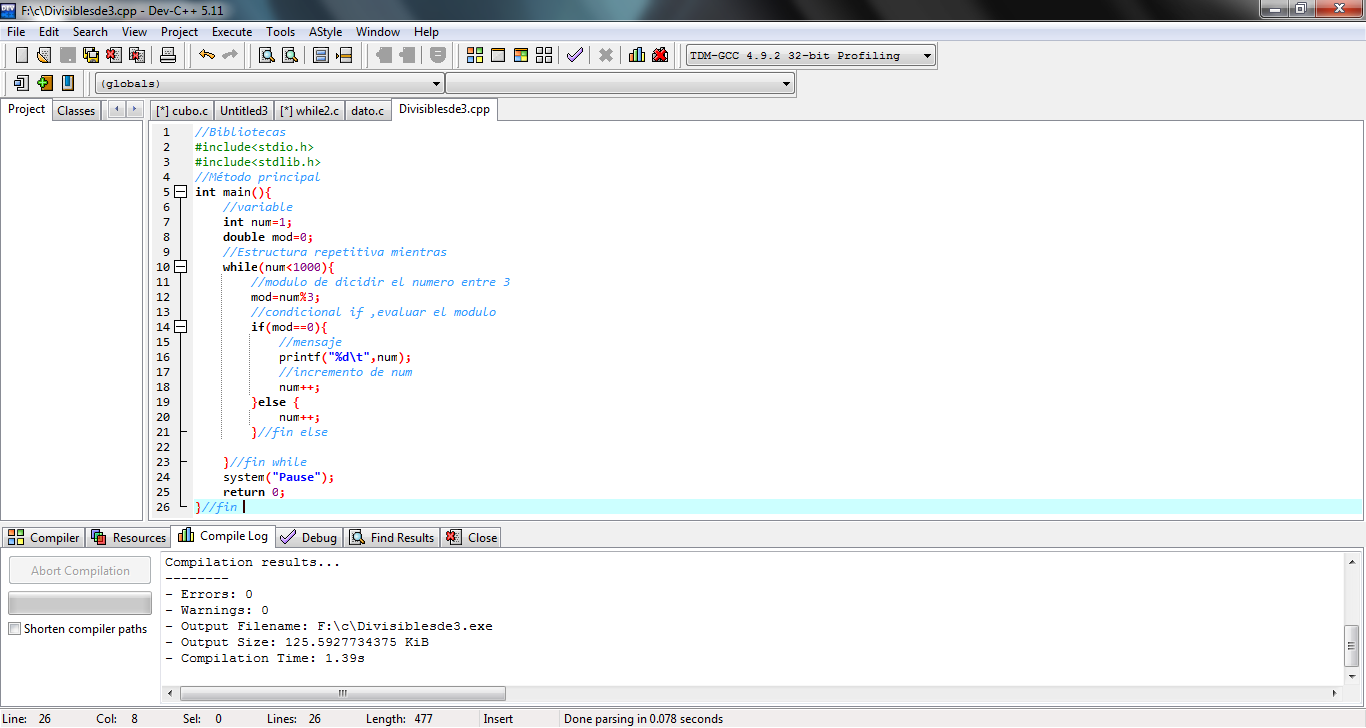
Imprimir los numero divisibles entre 3 mayores a 0 y menores a mil

Ejemplo:3,6,9,12,15,18,21,…..999

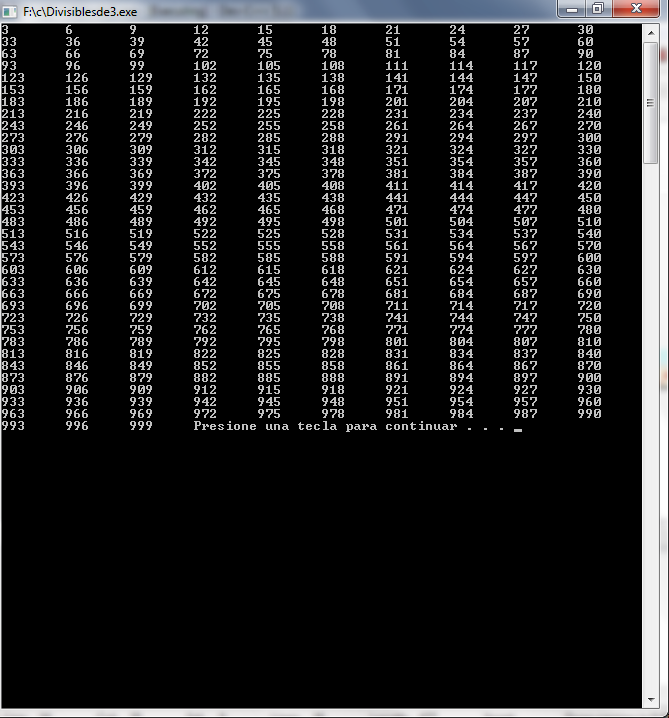
Cree un nuevo ´proyecto al cual nombre Divisiblesde3 con la extensión .c para que de esta manera se pueda generar el .exe al ejecutarlo



Una vez dentro del área de trabajo desarrolle el siguiente código, en el cual inicialice mi variable num en 1 sabiendo que este ya es un valor mayor a 0 como se indica en las instrucciones del ejercicio, posteriormente se evaluá la variable num para que se repitan las instrucciones dentro del ciclo mientras num sea menor a 1000, y para saber cuáles son los valores divisibles entre 3, obtenemos el residuo al ser dividido num entre 3, y por medio de un if evaluamos si el residuo es igual a 0 entonces imprime el valor que posee en ese momento la variable num y la incrementa y en caso contrario solo incrementa la variable pero, no imprime el valor.

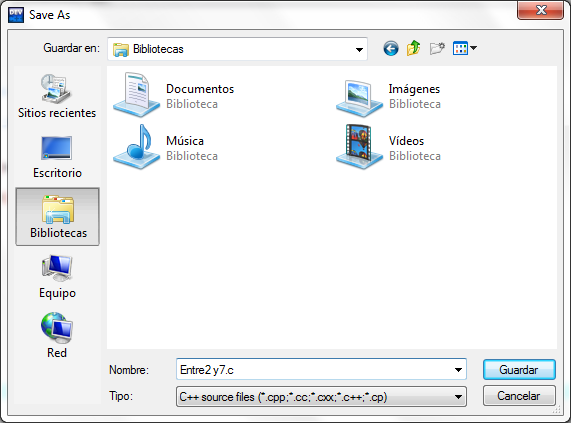


Al ejecutar el programa se muestran los siguientes valores en consola.

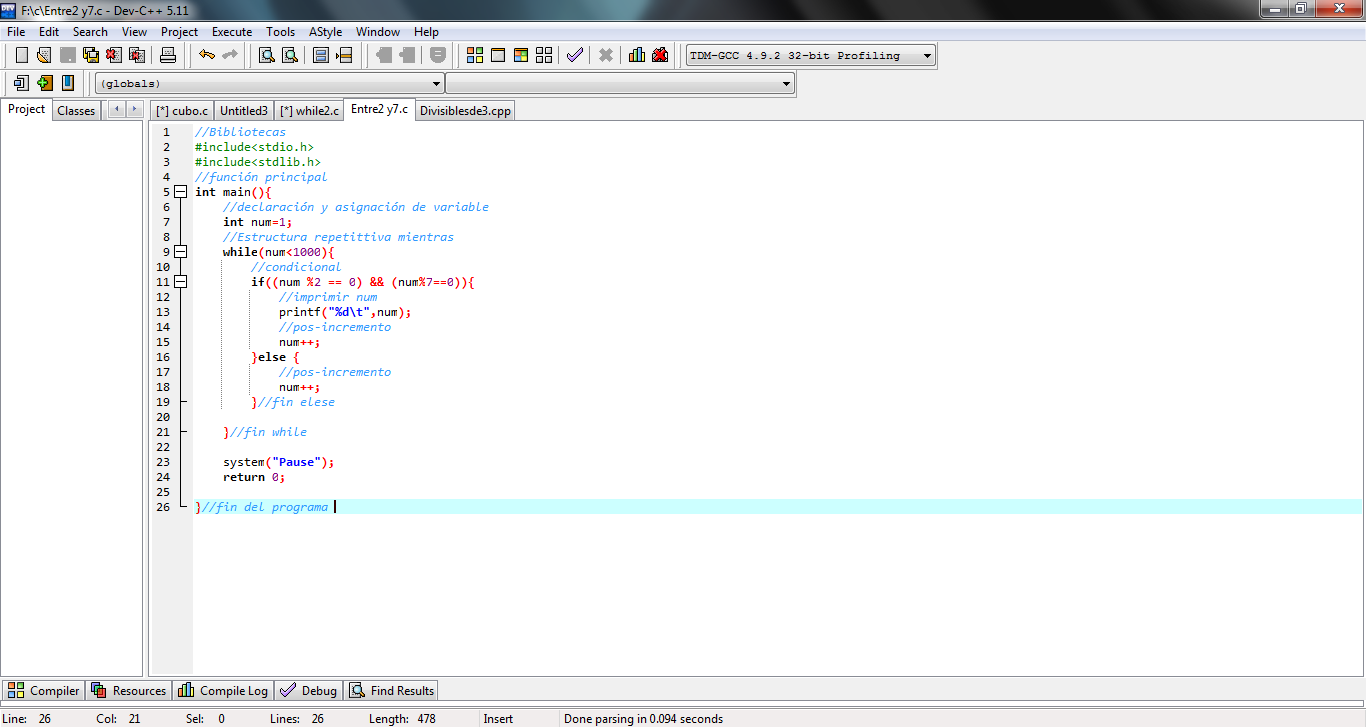


Imprimir los numero divisibles entre 2 y entre 7, mayores 0 y menores a mil

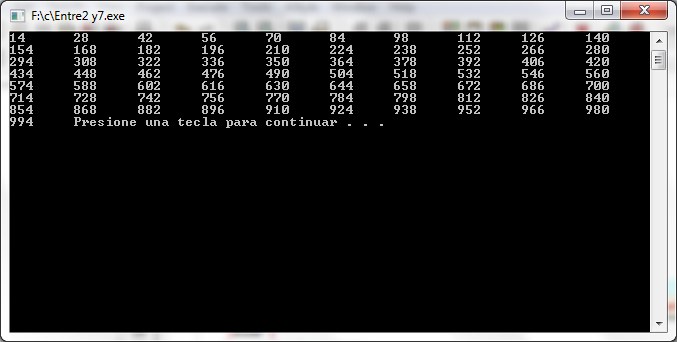
Al igual que el ejercicio anterior creo un programa con el siguiente nombre.



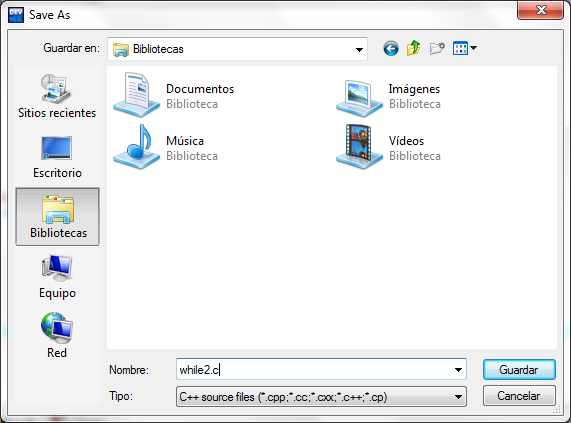
En este programa lo único que cambia a comparación del anterior es que ahora en el if se evalúa el resultado de los módulos obtenidos al dividir la variable num entre 2 y 7, y si el resultado es igual a 0 se imprime el valor de num, en caso contrario solo se incrementa num



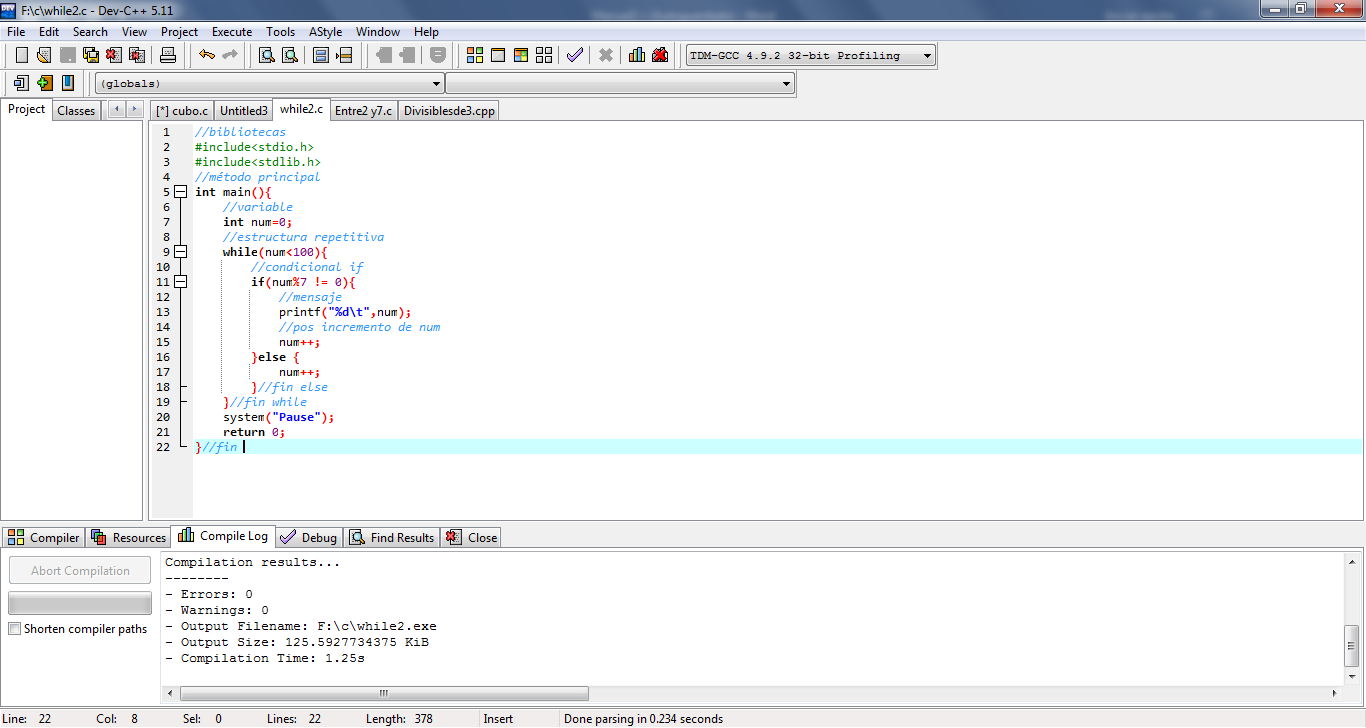
El resultado es el siguiente



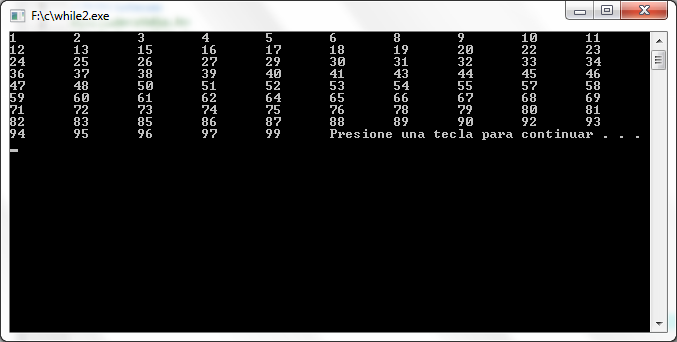
En este proyecto le asigne el siguiente nombre.



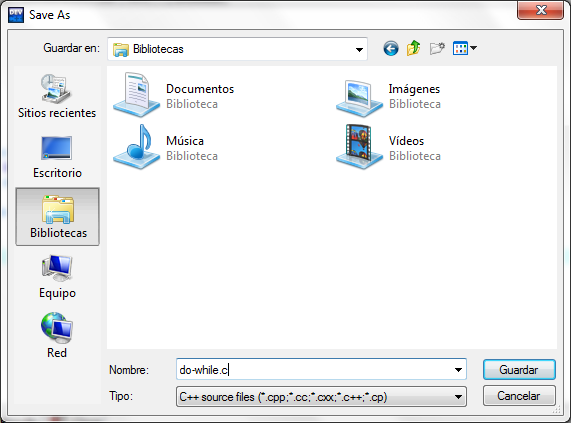
Este proyecto es similar a los anteriores puesto que se obtiene el módulo de los números que al ser divididos entre 7 sea igual a 0, por lo tanto, esto se realiza por medio de la condicional if, en cuanto el while repite este proceso mientras la variable num sea menor a 100, ya que cada vez que se recorre el ciclo se incrementa de uno en uno la variable num



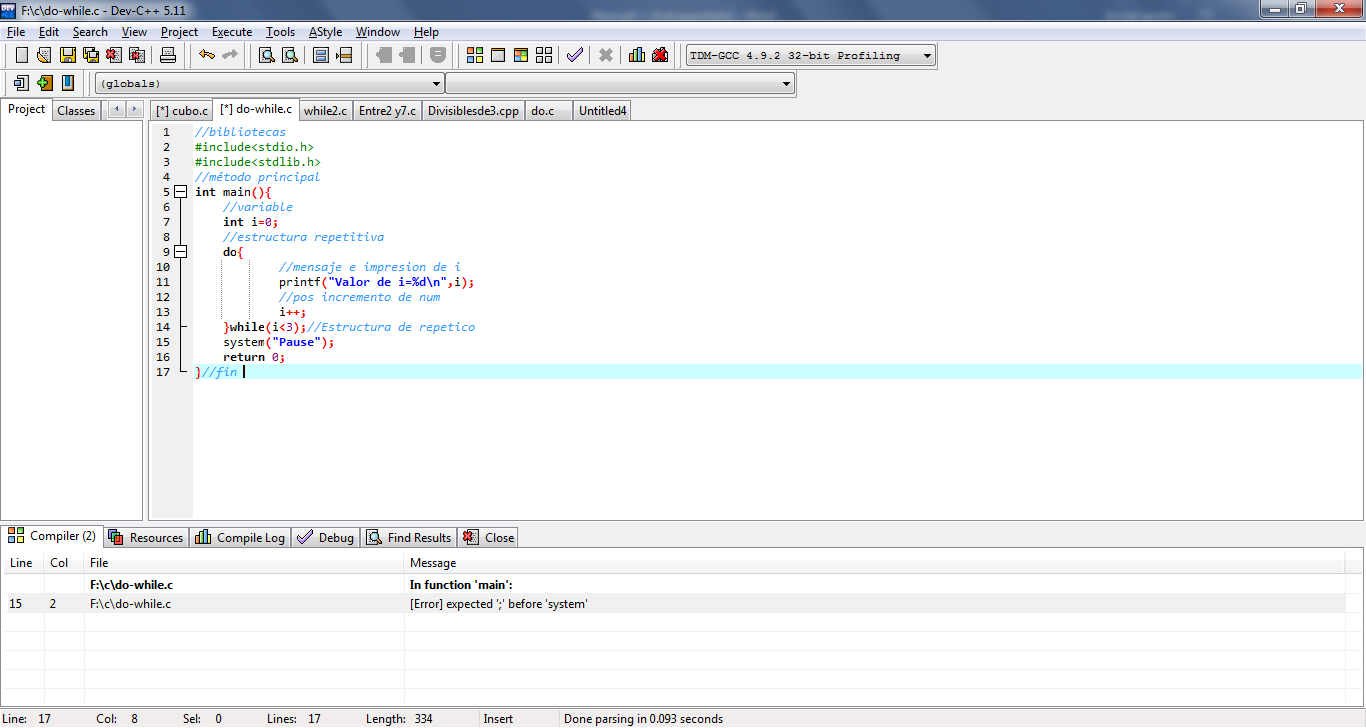
Finalmente, al ejecutar el programa estos son los valores que son múltiplos de 7



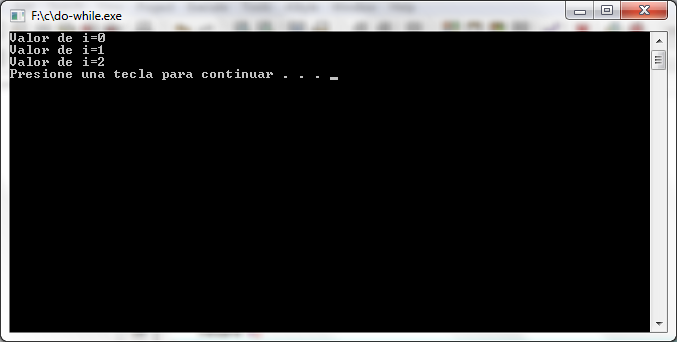
Este es el nombre del siguiente programa



La finalidad de este programa es analizar el funcionamiento del ciclo repetitivo do-while en el cual primero se ejecuta el ciclo y posteriormente se evalúa la variable num, por lo tanto, en el siguiente ejercicio si se imprime i tres veces puesto que el ciclo se ejecuta antes de ser evaluado.

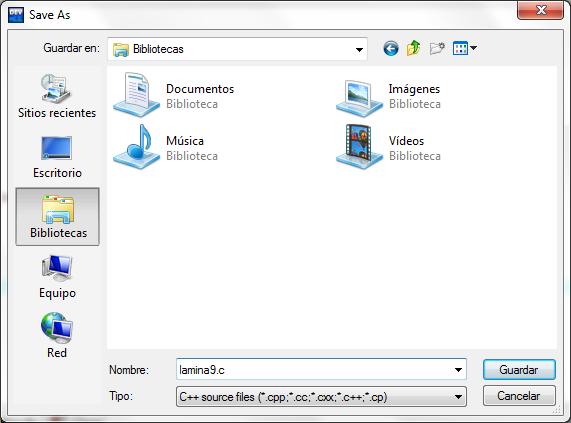


este es el resultado obtenido al mostrarse en consola

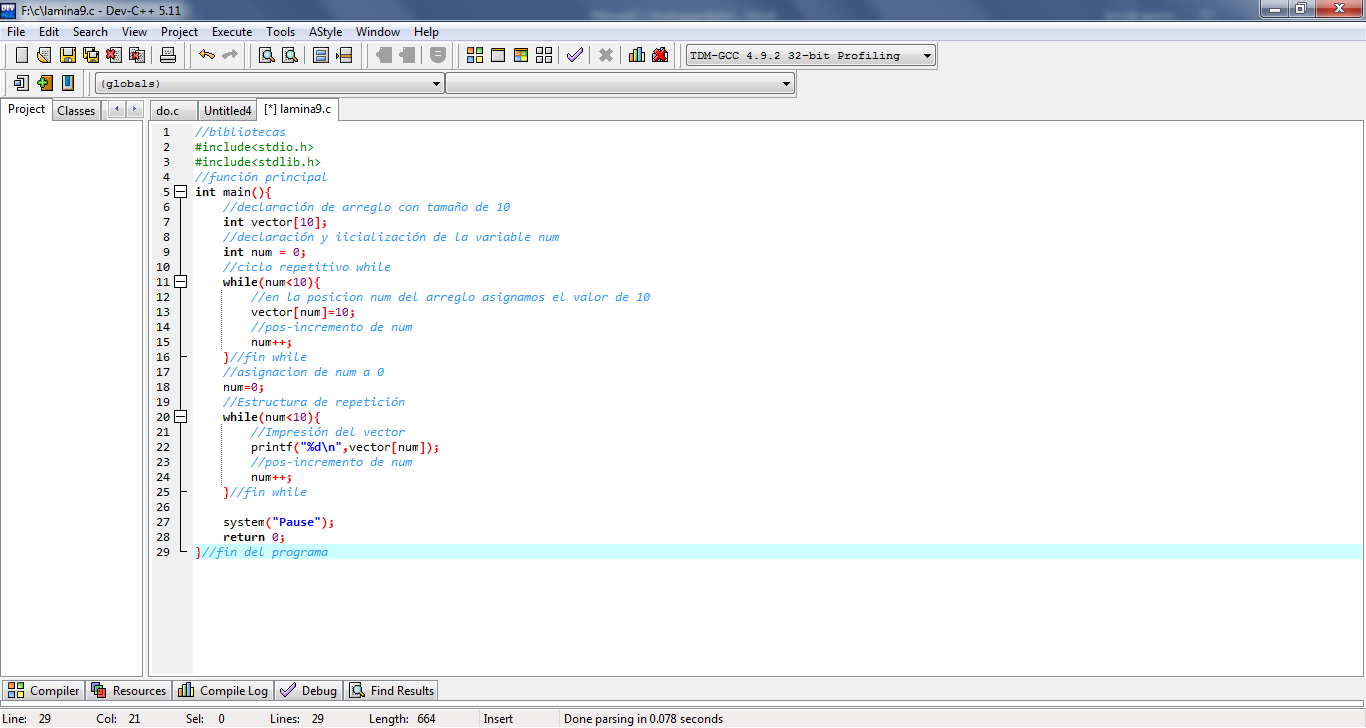


# Lamina 10 Arreglos

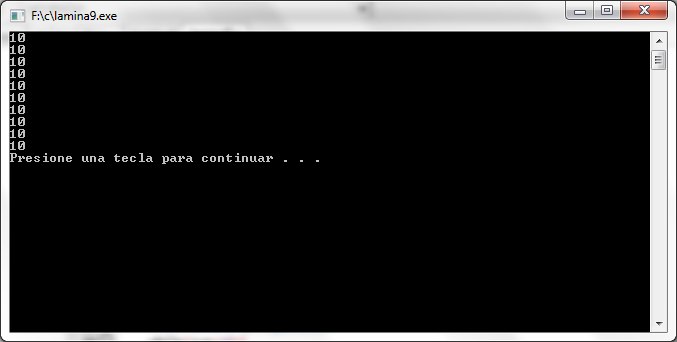
Creo un nuevo proyecto y lo guardo con el siguiente nombre, además de colocar la extensión .exe



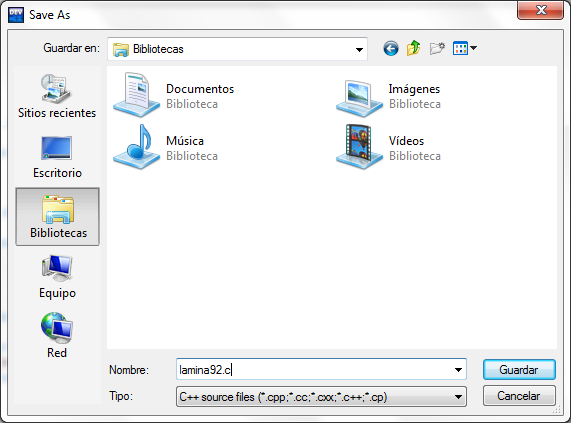
Desarrolle este código en el cual analizamos el funcionamiento de los arreglos, así como la sintaxis, lo primero que realice fue declarar mi arreglo y entre los corchetes asigne el tamaño además de indicar que este arreglo será de tipo int, por medio del primer ciclo while vamos recorriendo las posiciones del arreglo y asignándole el valor de 10 a cada una de sus posiciones, en cuanto al segundo siclo while imprime dichas posiciones del arreglo



De esta forma es como compila el proyecto y se ejecuta de acuerdo a las instrucciones programadas.



creo el siguiente proyecto, asignándole el siguiente nombre

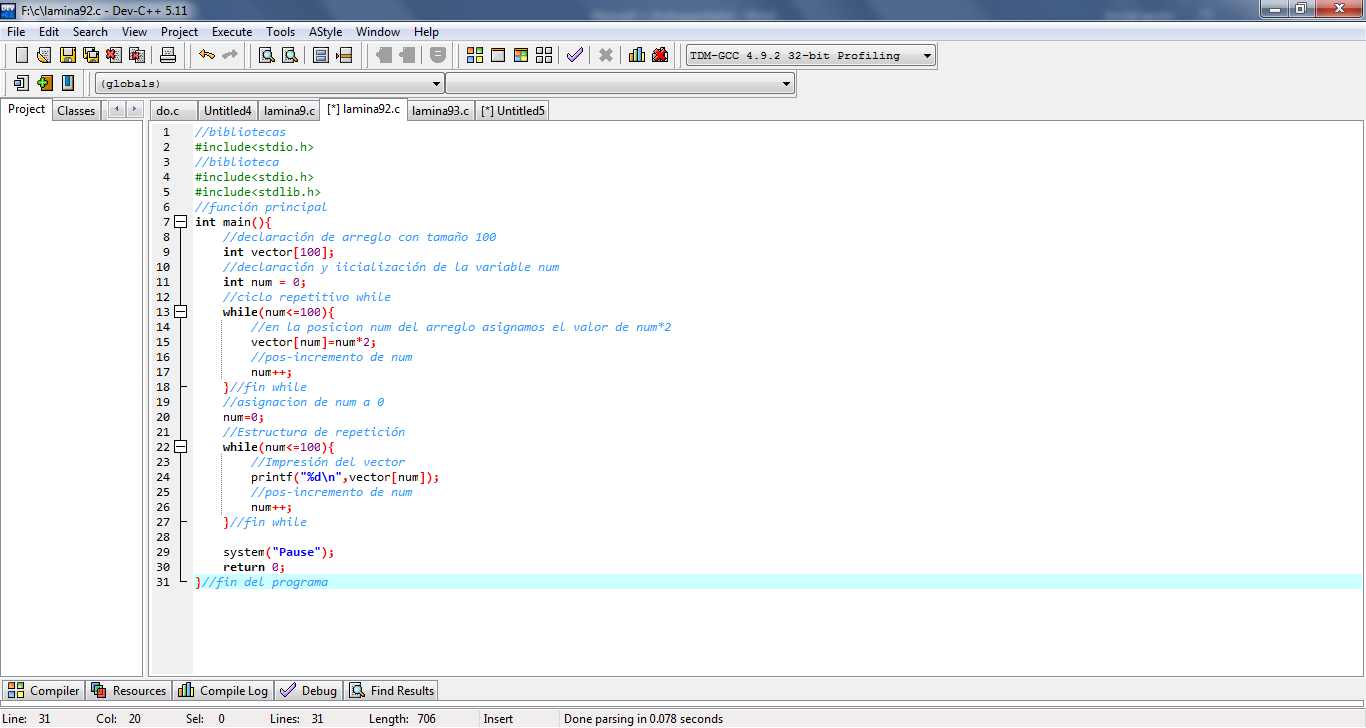


Crear un arreglo de 100 posiciones.

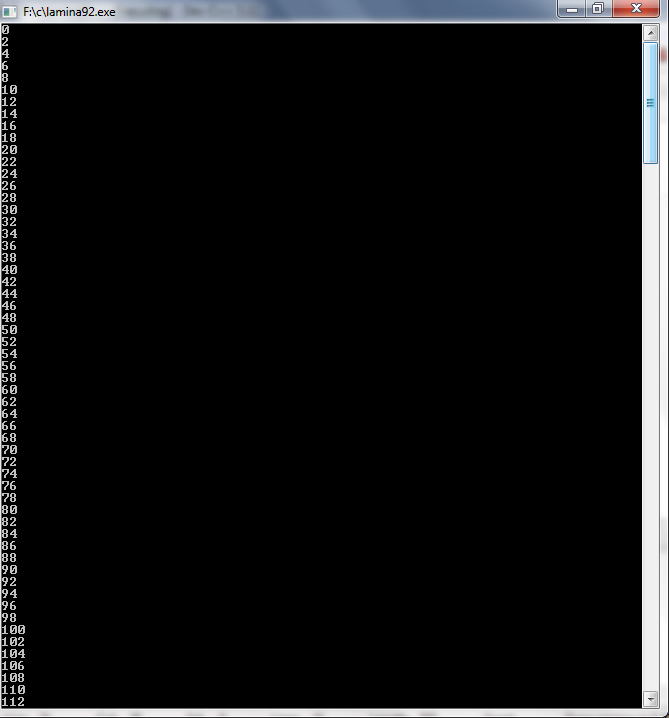
Llenar el arreglo con la tabla del 2.

Mostrar el arreglo en pantalla.

Primero asigno el tamaño de mi arreglo, posteriormente asigno el valor de 0 a mi variable num, en el primer ciclo asigno el valor de n\* dos a cada una de más posiciones, además de que esta variable va incrementando de uno en uno y en cuanto al segundo ciclo se imprime dicho vector



Este es el resultado obtenido en consola

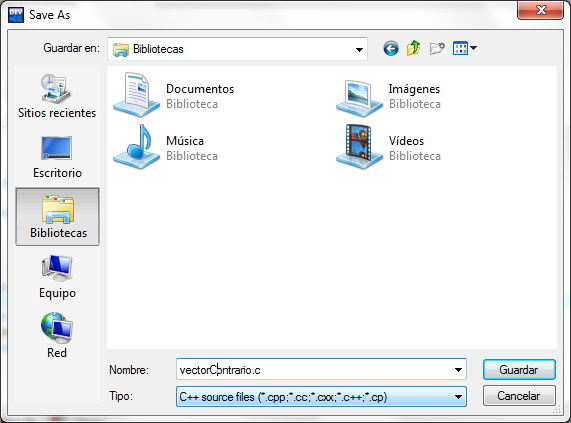


Crear un arreglo de 100 posiciones.

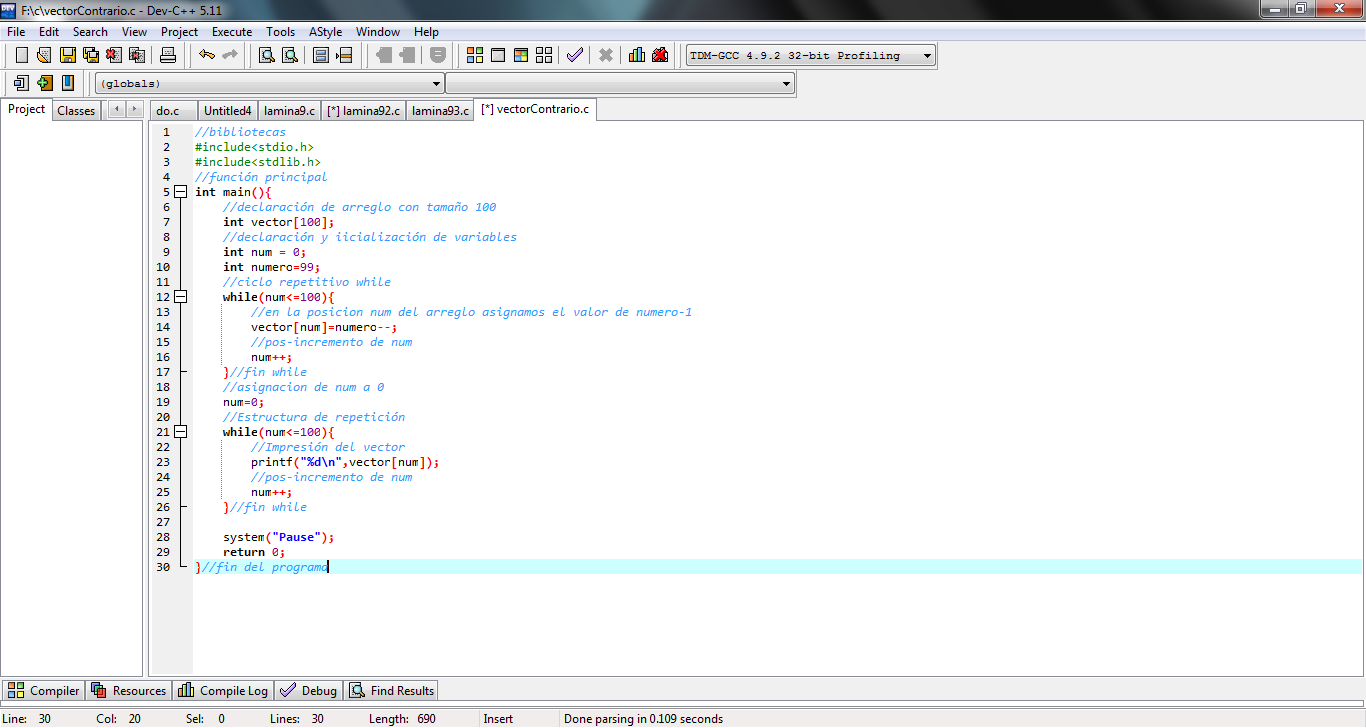
Llenar el orden inverso al índice.

Mostrar el arreglo en pantalla.

Creo mi proyecto como se ha realizado con los anteriores, nombrándolo de la siguiente manera



En este código me fue necesario crear una nueva variable para no alterar a num, puesto que esta variable se encarga de asignará las posiciones y numero de asignar el valor que posee cada posición, por ello esta variable va inicializada en 99 y este será el primer valor para la primera posición y al recorrer otra posición este valor va pos decrementando de 1 en uno y por ultimo con el segundo ciclo while se imprimen los valores que posee en arreglo en cada una de sus posiciones



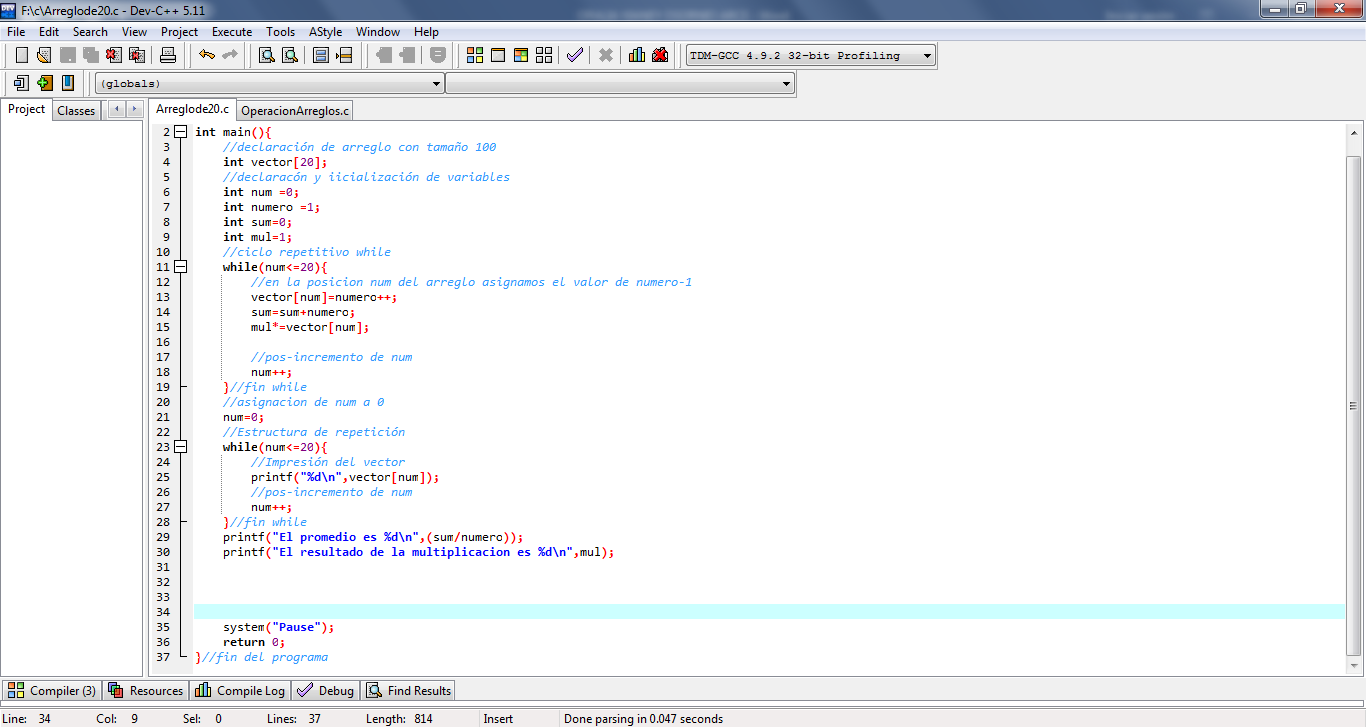
Este es el resultado obtenido, donde va disminuyendo el valor de cada vector según aumenta su posición



Creo un nuevo arreglo con el siguiente nombre, su nombre hace referencia a su tamaño



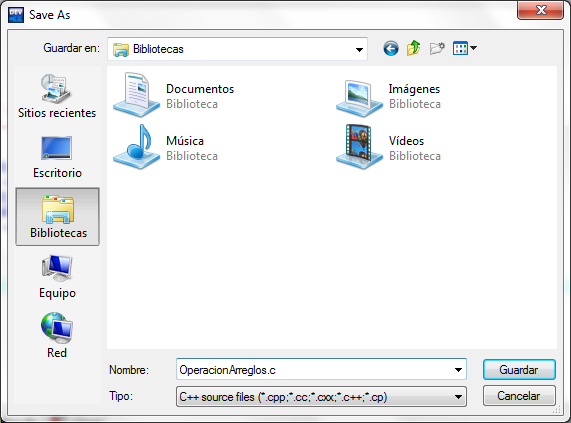
Este programa consiste en llenar el arreglo de números distintos, por lo tanto yo o llene con valores según el incremento de la variable numero, posteriormente por medio de (sum=sum + numero;) sumo todos los valores asignados a cada posición del arreglo y después los divido entre el numero de posiciones para así obtener el promedio



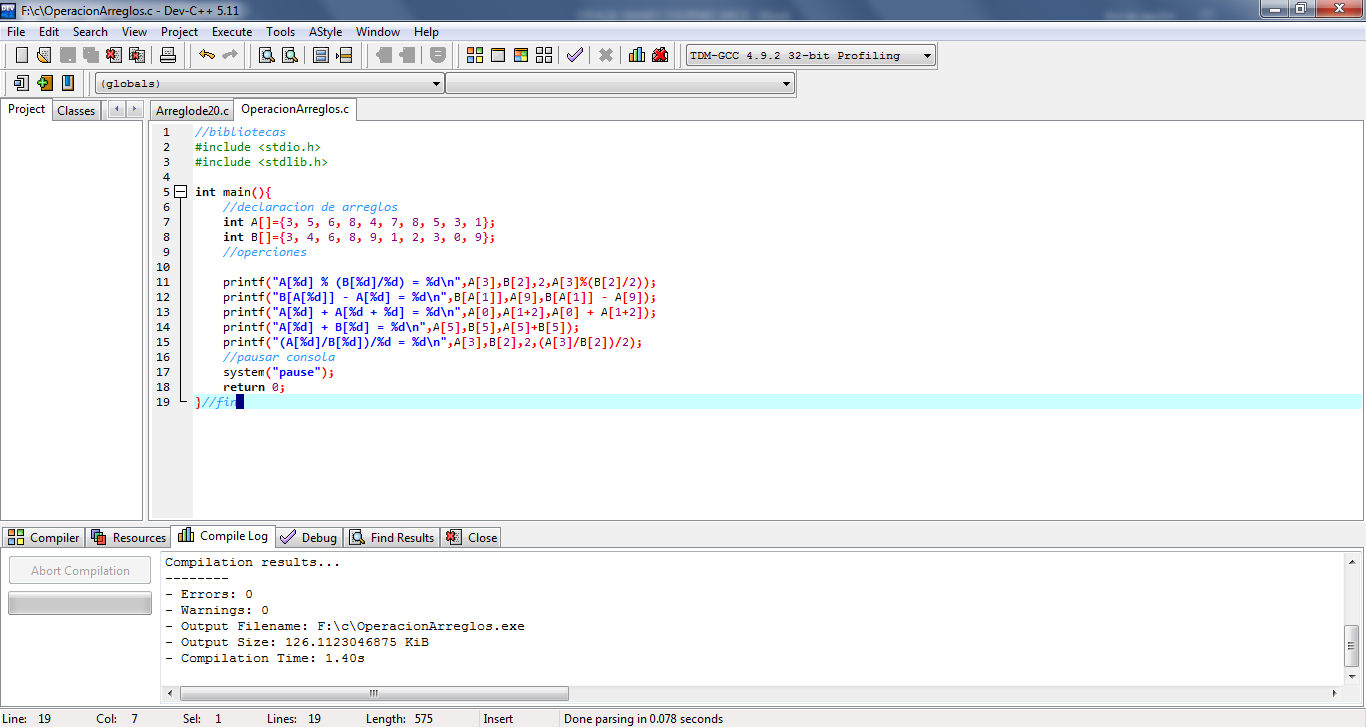
Al ejecutar este ejercicio se muestra el siguiente resultado en pantalla



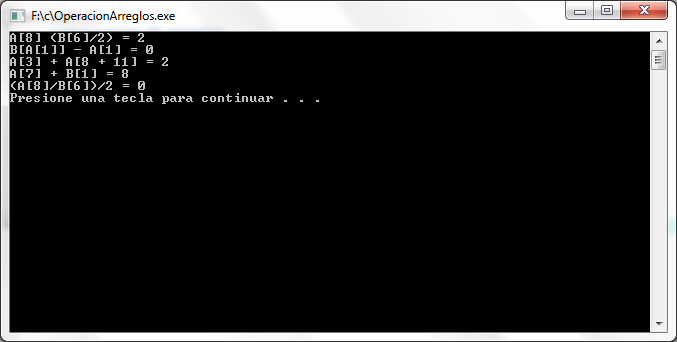
Finalmente creo un nuevo programa con el siguiente nombre.



En el cual realice las siguientes operaciones mandando a imprimir cada una de ellas en pantalla

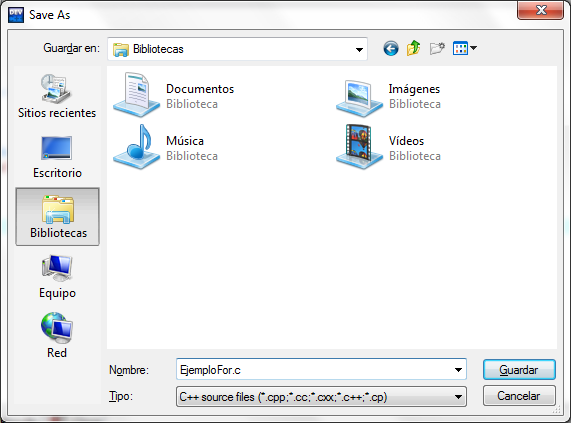


Siendo este el resultado al ejecutar

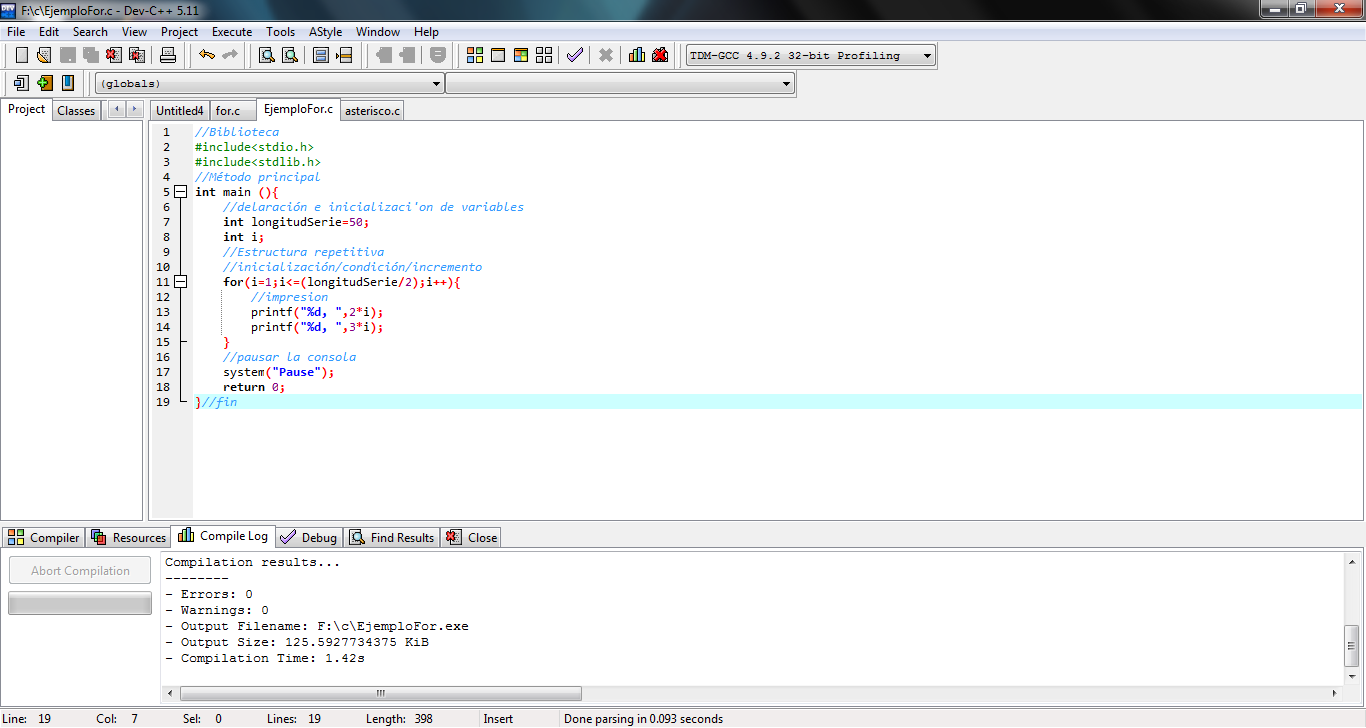


# Lamina11 For

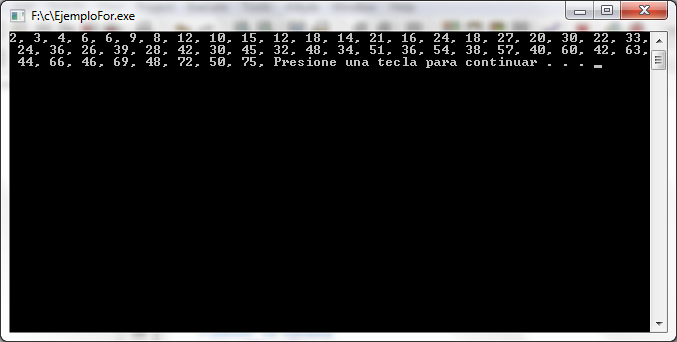
Cree el siguiente proyecto, al cual nombre como se muestra en la captura



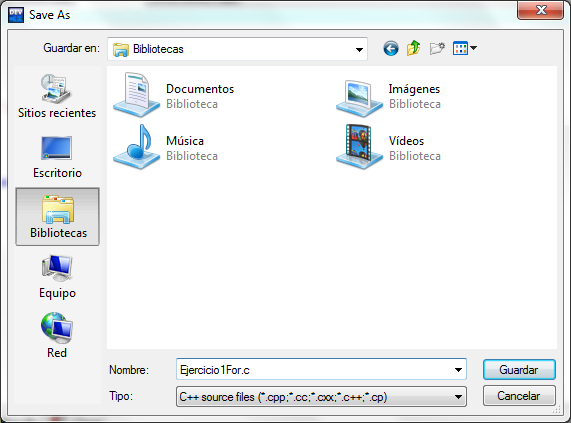
En este primer programa del ciclo for solo basta con declarar nuestras variables o iniciarlas, sin embargo el for es similar a el while, pero a diferencia de los demás ciclos este permite que se inicialice la variable, que se declare la condición para evaluar hasta cuándo debe dejara de repetirse el ciclo y finalmente el incremento o decremento de la variable según sea el caso, en este programa se inicializa la variable i en 1 y este ciclo se repetirá mientras i sea menor o igual a 50/2 puesto que este es el valor que posee la variable longitudSerie y dentro del ciclo se muestran las instrucciones a ejecutar, en las cuales se imprimen las operaciones de 2 por el incremento de i y 3 por el incremento de i y finaliza el ciclo cuando la condición no se cumpla



Este es el resultado generado por el programa al mostrarse en pantalla

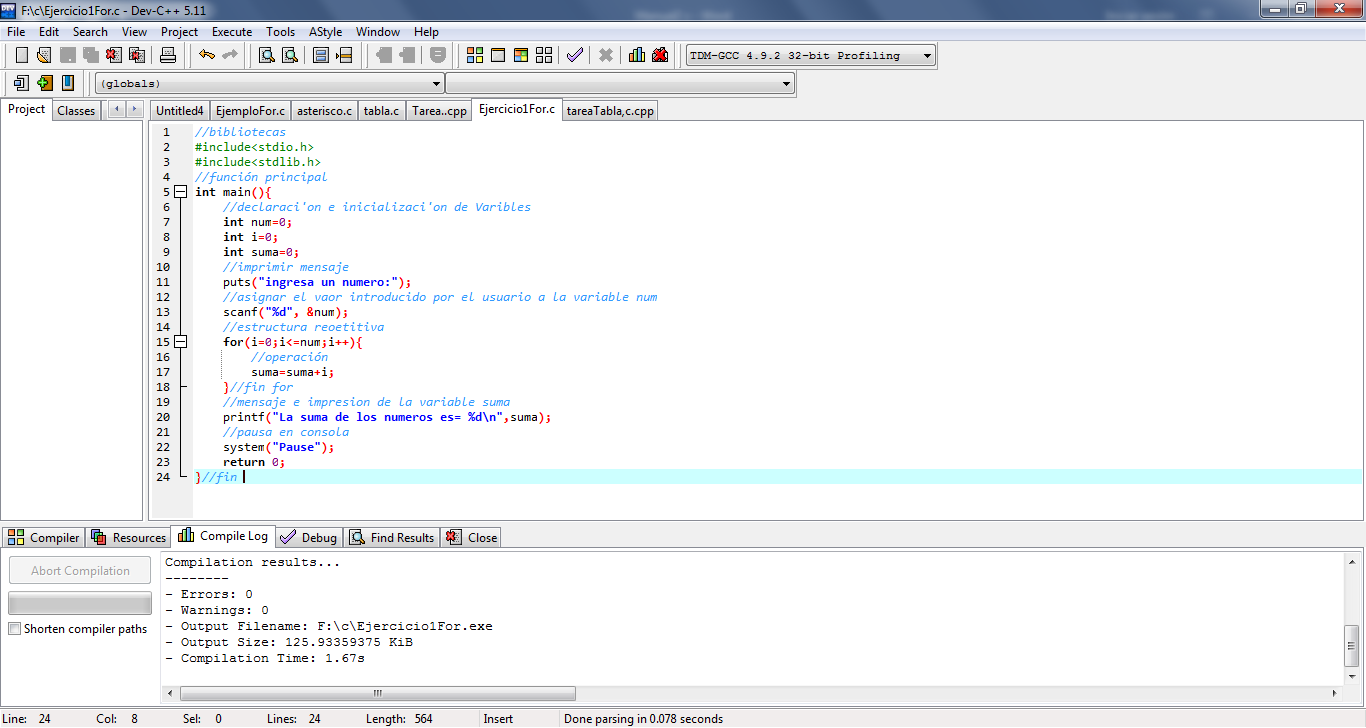


Creo un nuevo proyecto y lo nombre de la siguiente manera

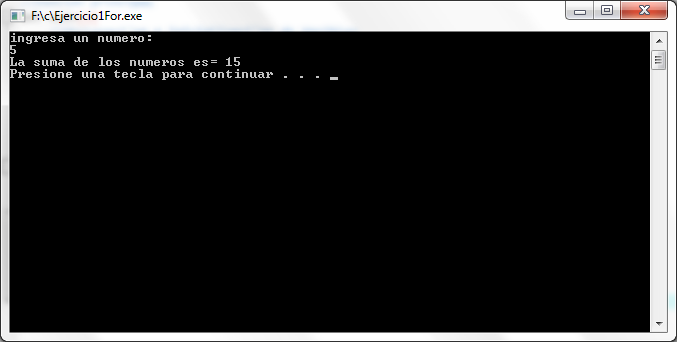


Escribe un programa que reciba un número N del usuario y haga la suma de todos los números desde 1 hasta N. Ej. >> 5 1+2+3+4+5 = 15

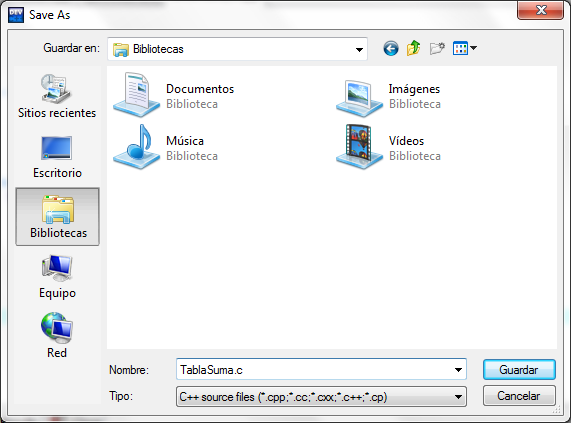
En este programa dentro del ciclo for solo realice una operación en la cual vamos sumando los valores de i cada vez que incrementa y fuera del ciclo repetitivo mande a imprimir un mensaje indicando que era el valor de la suma total de todos los valores de i

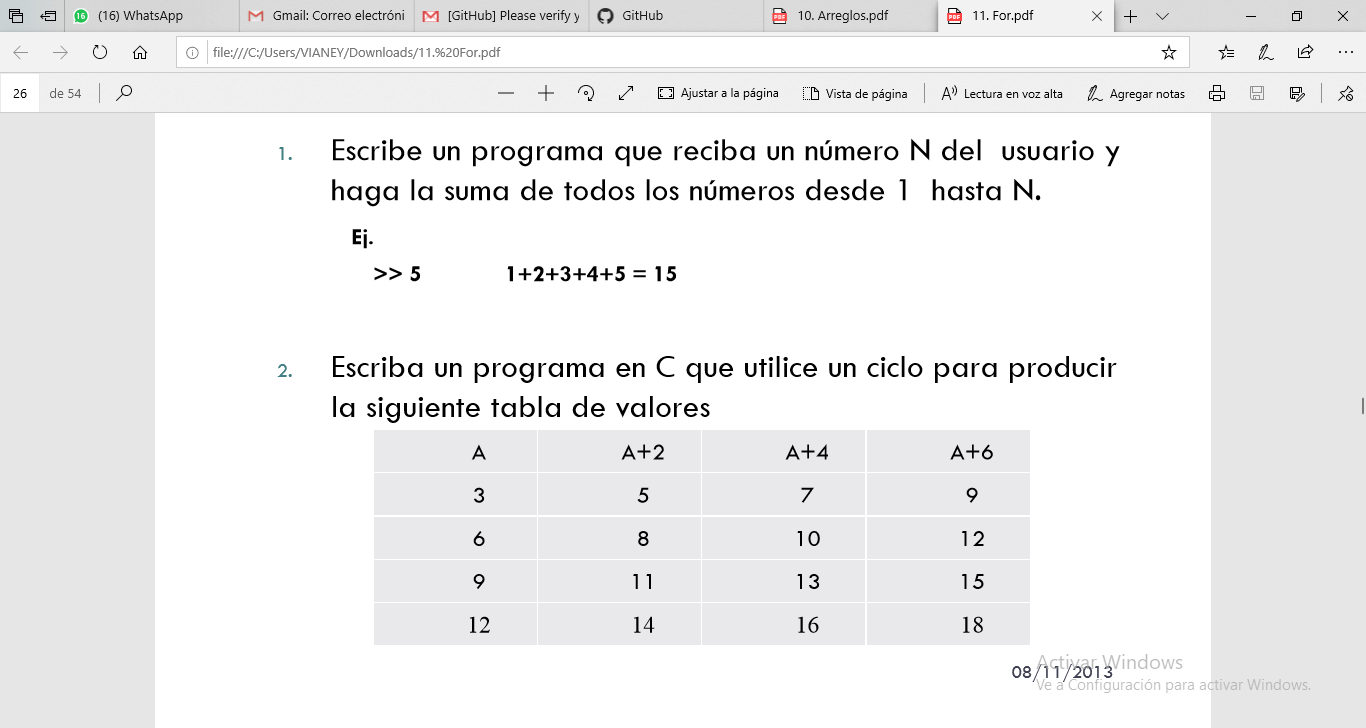


Al ingresar el calor en consola, se obtiene el factor de dicho número, puesto que se suman los valores anteriores al valor ingresado, así como el valor que se ingresó.

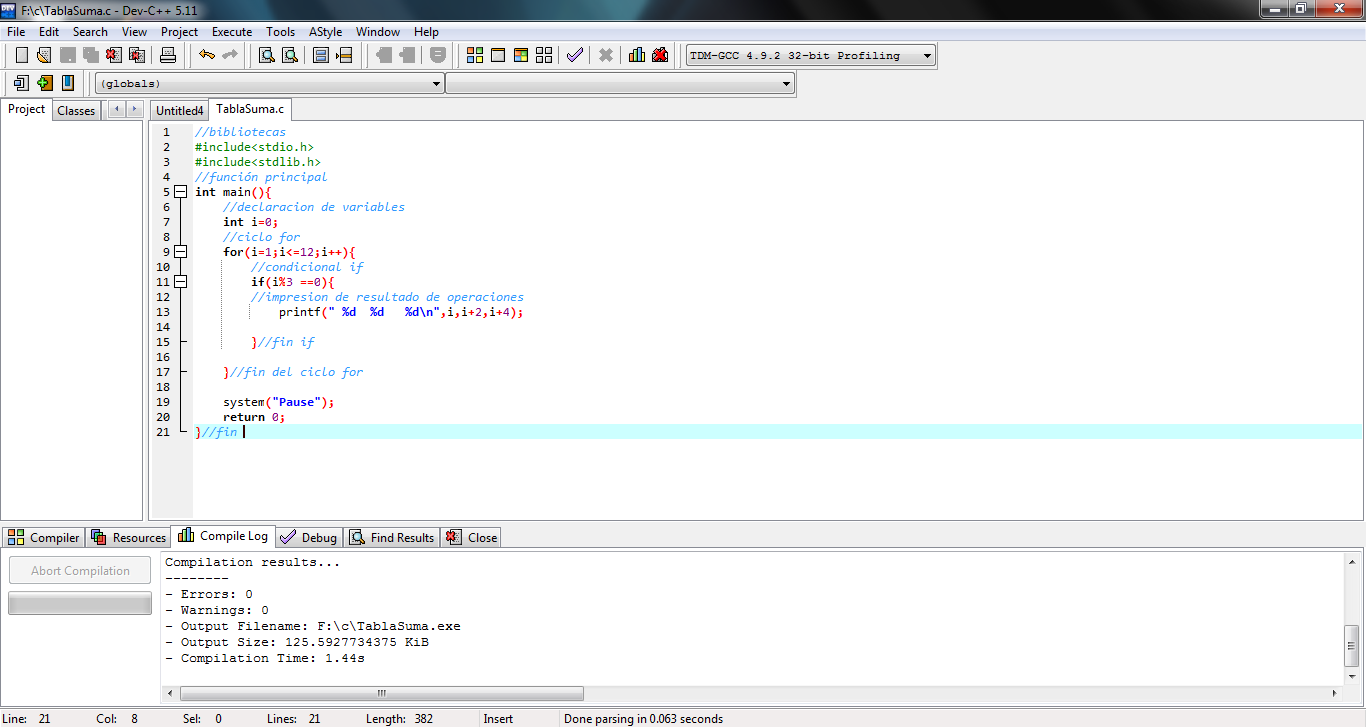


Escriba un programa en C que utilice un ciclo para producir la siguiente tabla de valores.

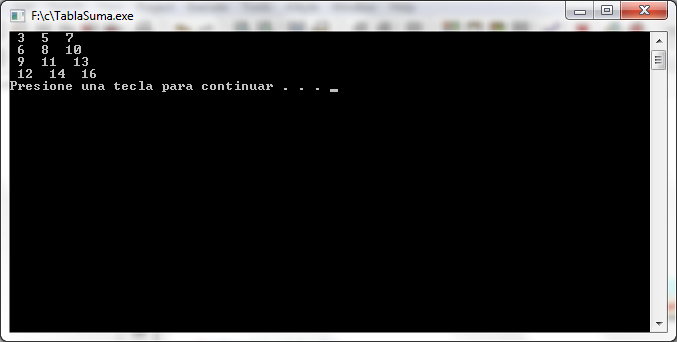


 Escriba un programa en C que utilice un ciclo para producir la siguiente tabla de valores

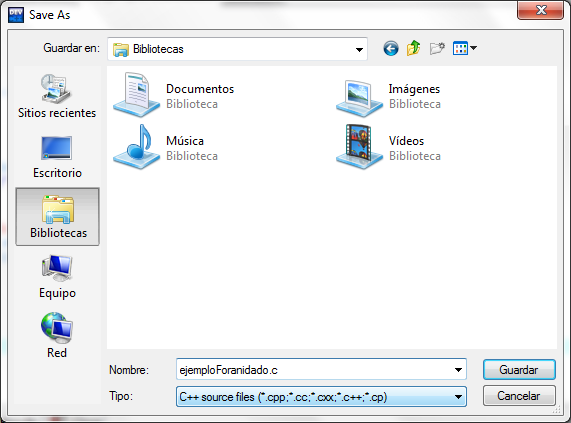
En este programa el ciclo for se va a ejecutar mientras el valor de i sea menor o igual a 12, mientras que i vale 1 y va incrementándolo de 1 en 1, mientras que al ejecutarse el ciclo se imprimen las operaciones correspondientes primero i mas 2 y después más 4, como se muestra en esta captura



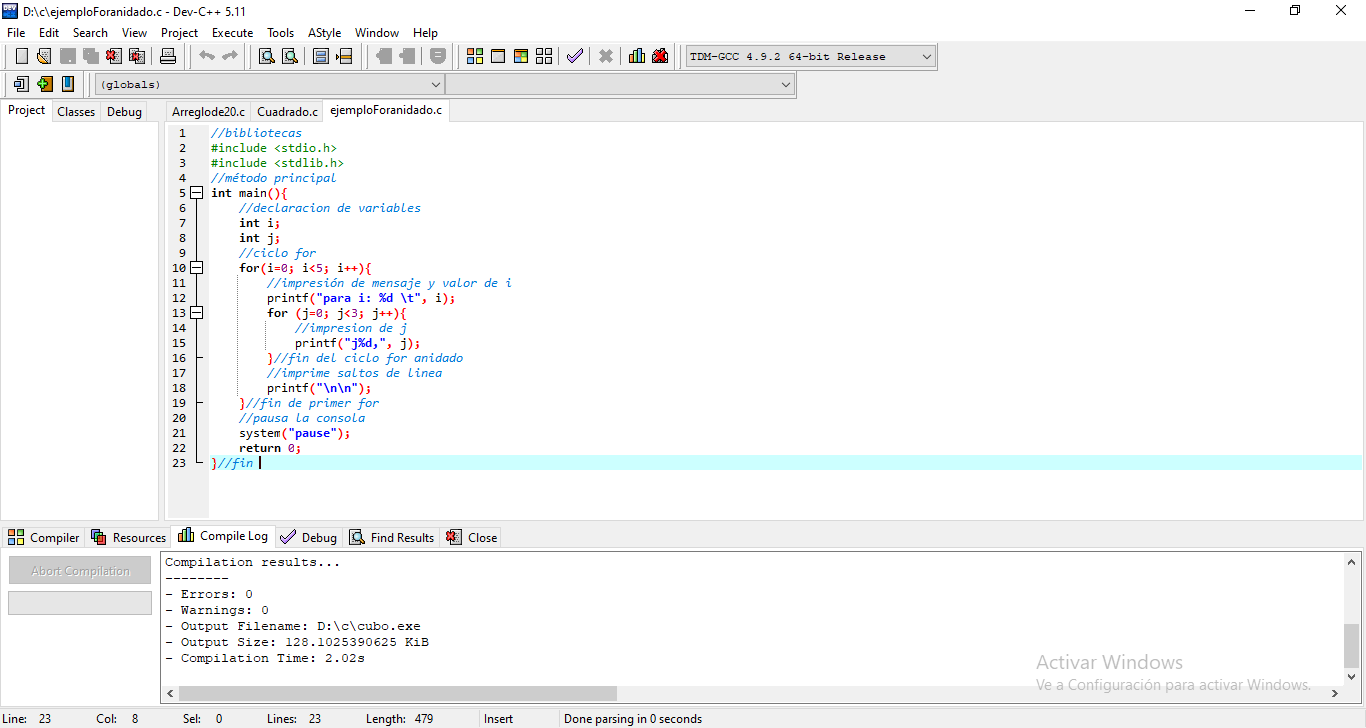
Este es el resultado una vez que se muestra en consola



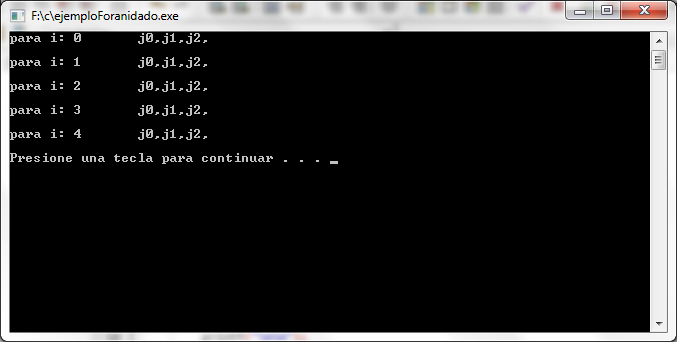
Guardo este nuevo proyecto con este nombre



La función del primer for es repetir el segundo ciclo for mientras que i sea menor a 5, por lo tanto, el ciclo for que se encuentra dentro del se repetirá 4 veces, mientras que este a su vez imprime sus instrucciones dentro del mientras que j sea menor a 3, tomando en cuenta que j es inicializada en 0

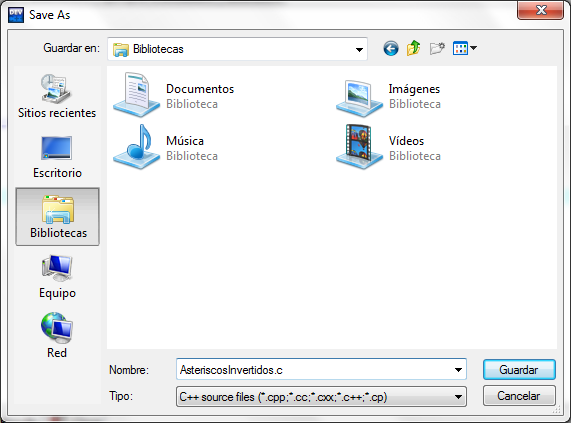


Este es el valor que se muestra en consola

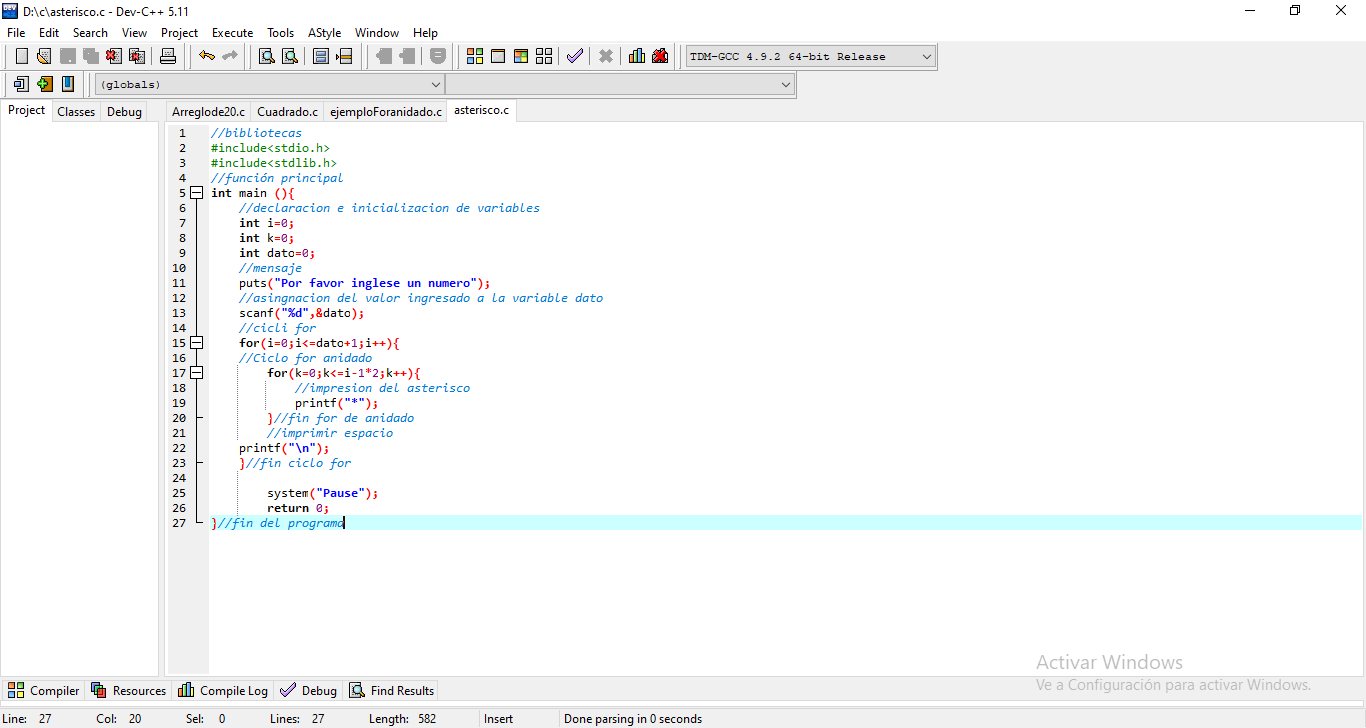


Escribe un programa que reciba un número entero N del usuario e imprima una escalinata de N pisos de asteriscos

Nombro mi proyecto de la siguiente manera



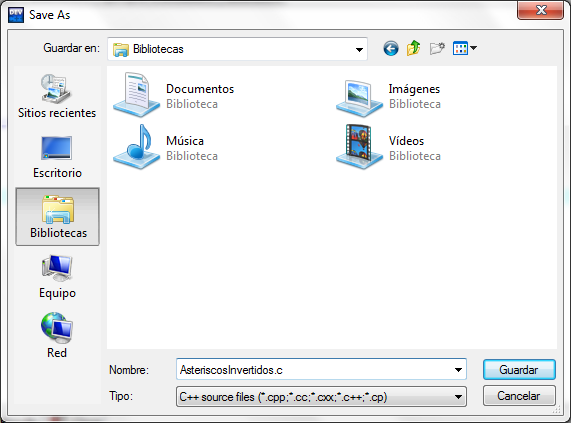
Este programa solicita un número y este es tomado como parámetro para el primer for el cual indica el número de veces que se repetirá el for que se encuentra dentro de él que es el que se encarga de imprimir los asteriscos según del valor de 1

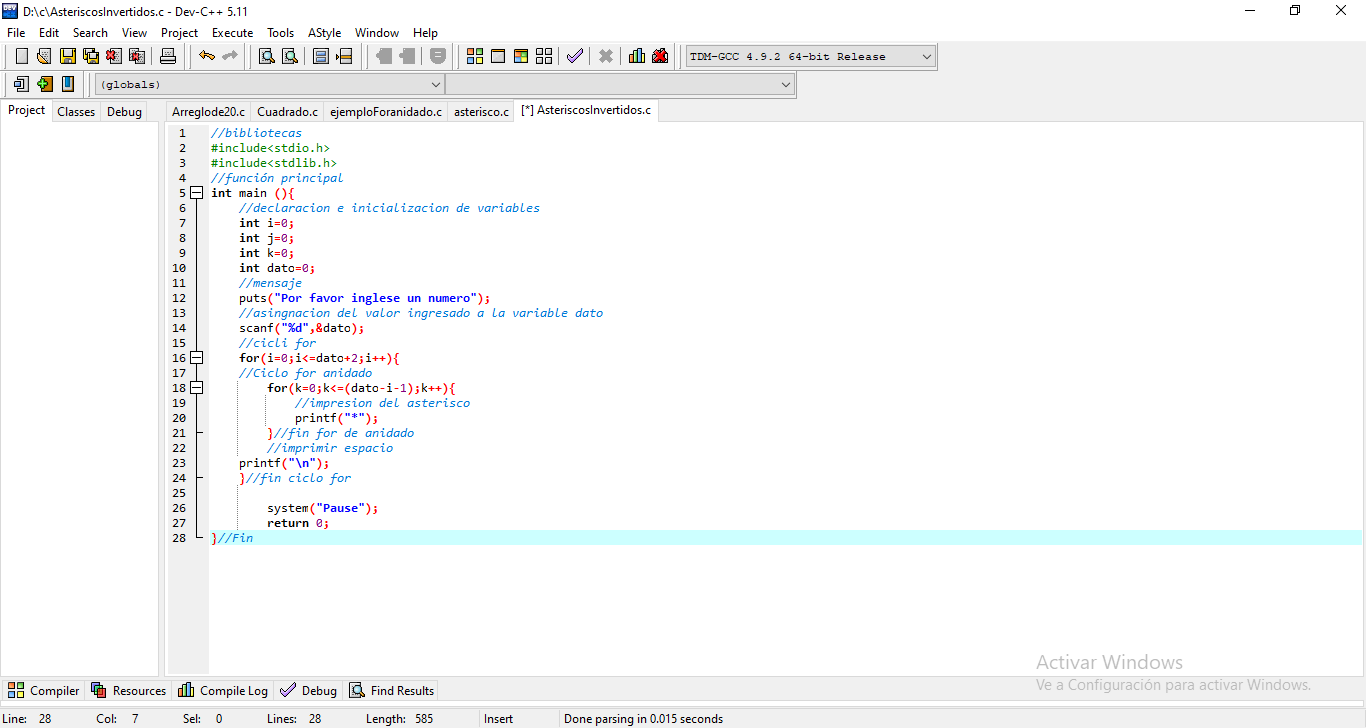


Este es el resultado una vez que se ingresa el valor en consola

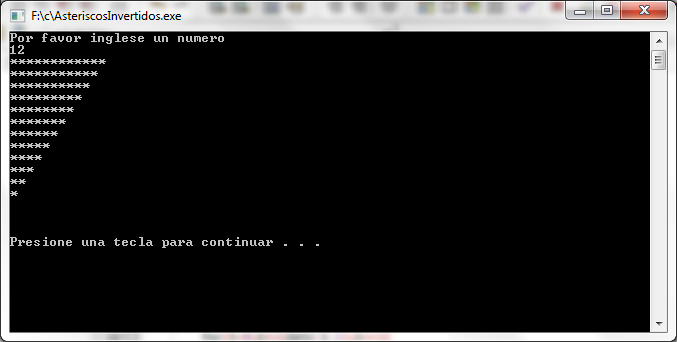


En cuanto al segundo programa lo llame de la siguiente manera



En este for solo cambio la condición del for que se encuentra anidado, puesto que lo único que se realiza es

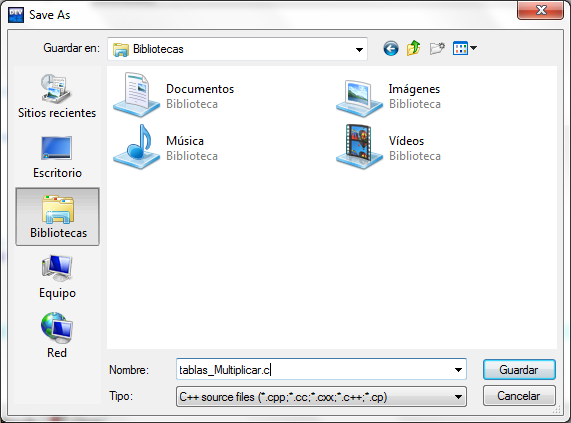
Este es el resultado que se muestra en consola



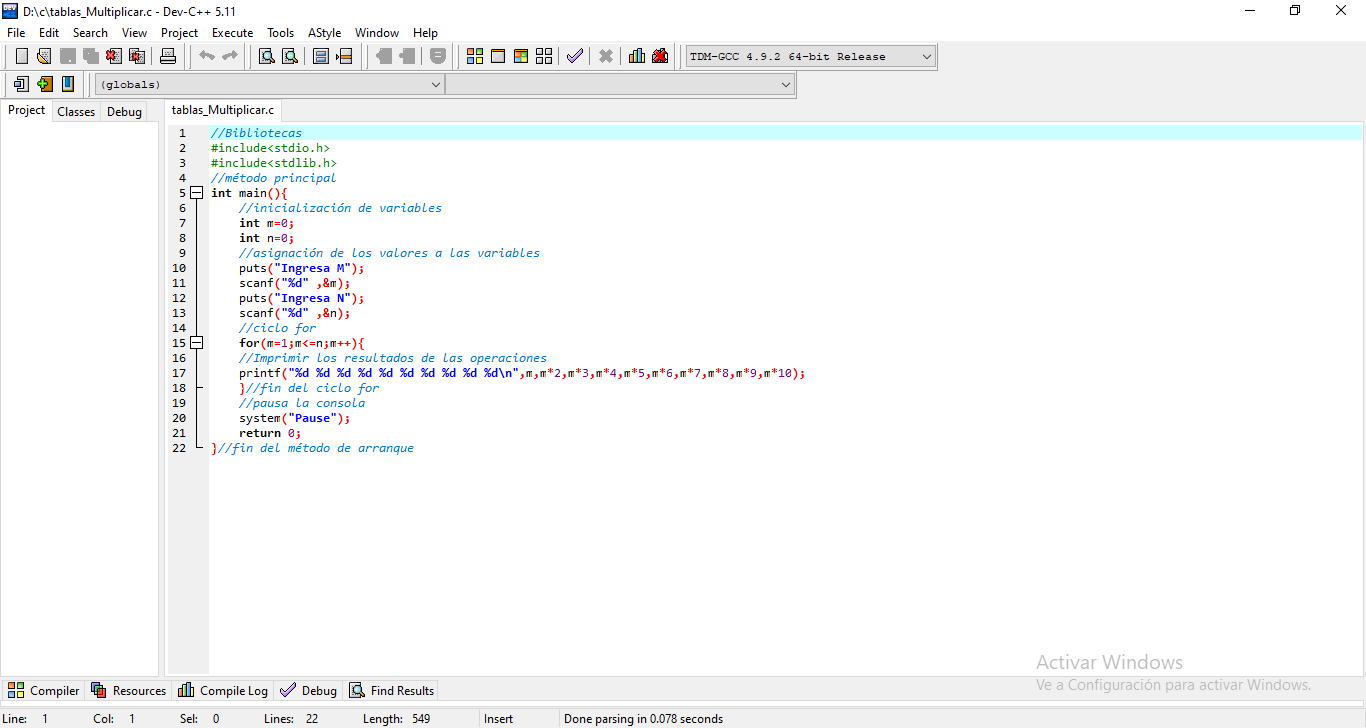
Haz un programa que muestre las tablas de multiplicar como se muestra a continuación.

 M y N son ingresadas por el usuario

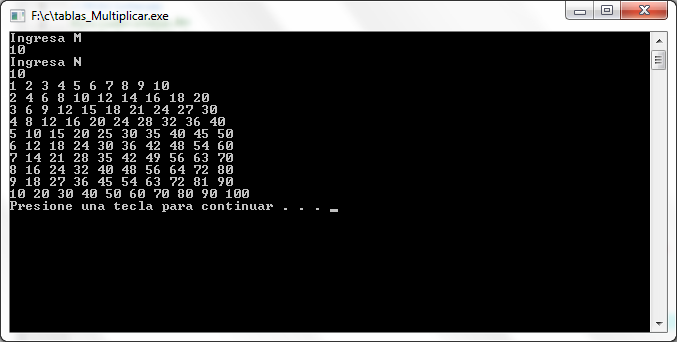
Este último programa lo llame de la siguiente manera



En este código lo primero que se realiza es obtener el valor de M y N por medio de un mensaje y lo asignamos a sus respectivas variables y mediante el for inicializamos a m la cual se repetirá mientras sea menor o igual a n que es el numero ingresado por el usuario, mientras que dentro del ciclo la instrucción a repetir son los valores de la multiplicación según sea de 1 a 10, sabiendo que i incrementa de uno en uno



Este es el resultado que se muestra en pantalla una vez que se ejecuta y se ingresan los valores de M y N



**V. Conclusiones:**

Mediante desarrollo de la práctica, retroalimentamos el análisis del funcionamiento de las condiciónales, ciclos de repetición simples y anidados así como las sintaxis de cada uno de ellos, con la finalidad de desarrollar diferentes aplicaciones arraigando los temas anteriormente vistos, logrando aplicar lógica matemática, como lo es en las operaciones tanto de arreglos como de valores de variables declaradas internamente o las introducidas por el usuario, del mimo modo logramos aplicar nuevas técnicas para la reutilización de código y reducción de líneas del mismo por medio de los ciclos