|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre de la práctica** | **PROGRAMAS C++** | | | **No.** | **1** |
| **Asignatura:** | **METODOS NUMERICOS** | **Carrera:** | **ING. SISTEMAS COMPUTACIONALES** | **Duración de la práctica (Hrs)** | **10** |

**I. Competencia(s) específica(s):**

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

Otro

**III. Material empleado:**

Dev C++

**IV. Desarrollo de la práctica:**

\*LENGUAJE C

-C es un lenguaje de programación estructurada desarrollado a principios de los años 70.

-Es un lenguaje de nivel medio, pero con muchas características de bajo nivel.

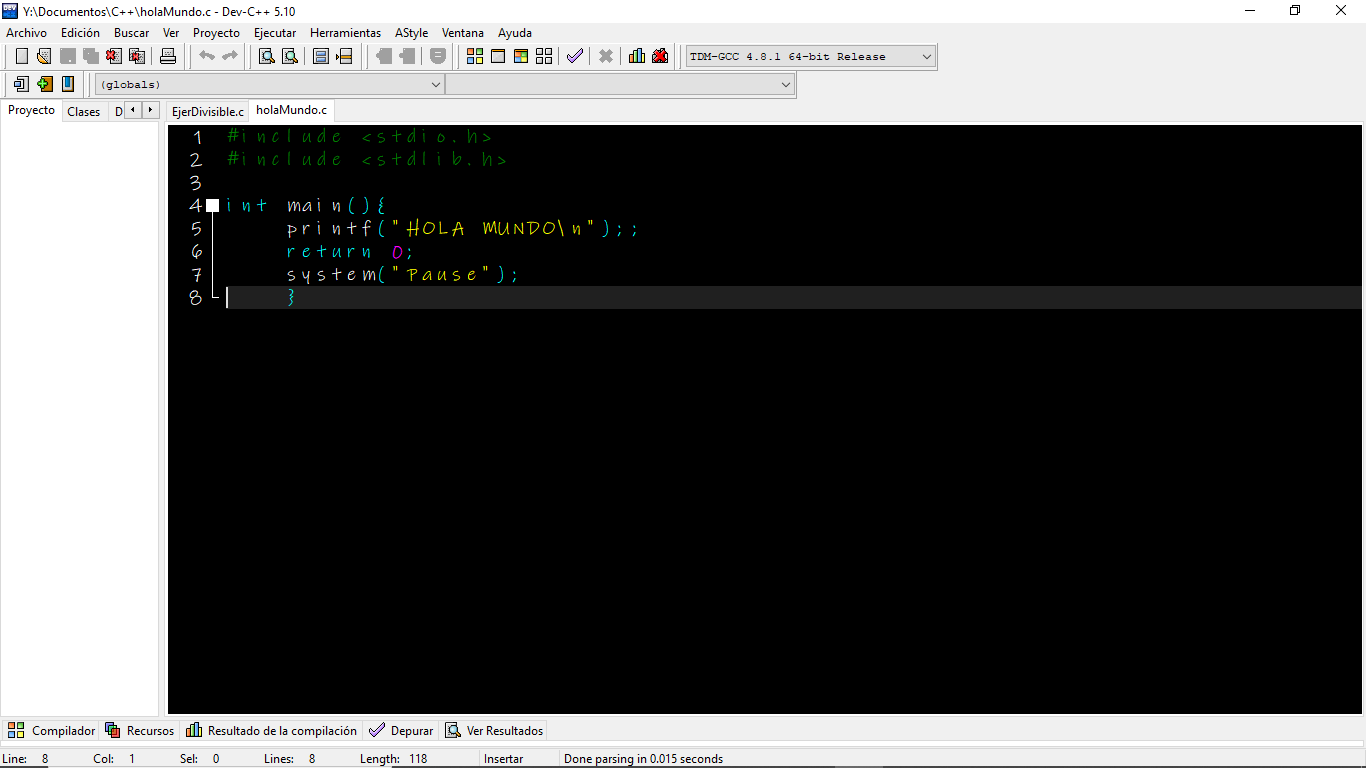
-Está disponible en un amplio abanico de plataformas.

Ejemplo 1:

#include <stdio.h> //Estandar de entradas y salidas

#include <stilib.h> //Estandar de bibliotecas

En este primer codigo mandaremos un mensaje “HOLA MUNDO”



Printf (print formated)

Sirve para imprimir en pantalla una cadena de caracteres con formato.

-Hay algunos caracteres especiales que no son imprimibles en pantalla como:

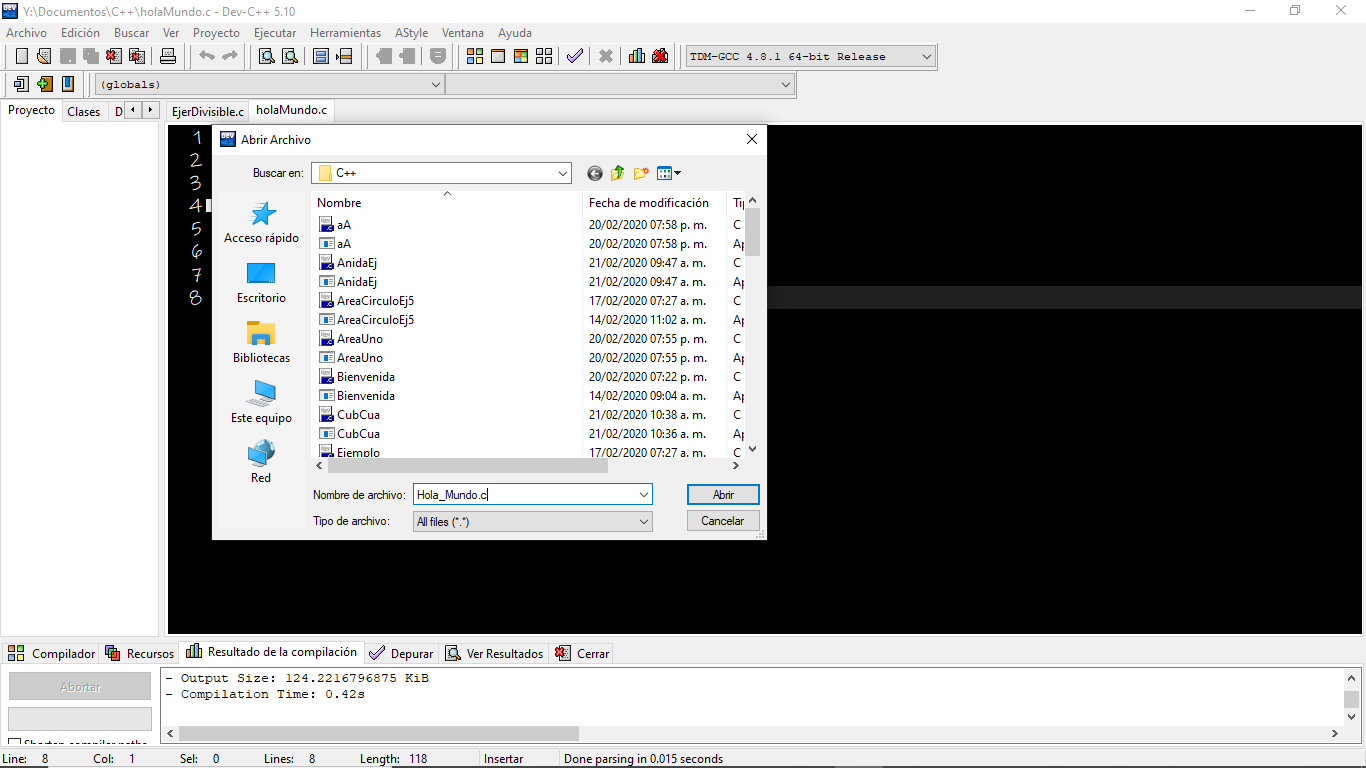
**\n** Salto de línea

**\t** Imprime un espacio de tabulador horizontal

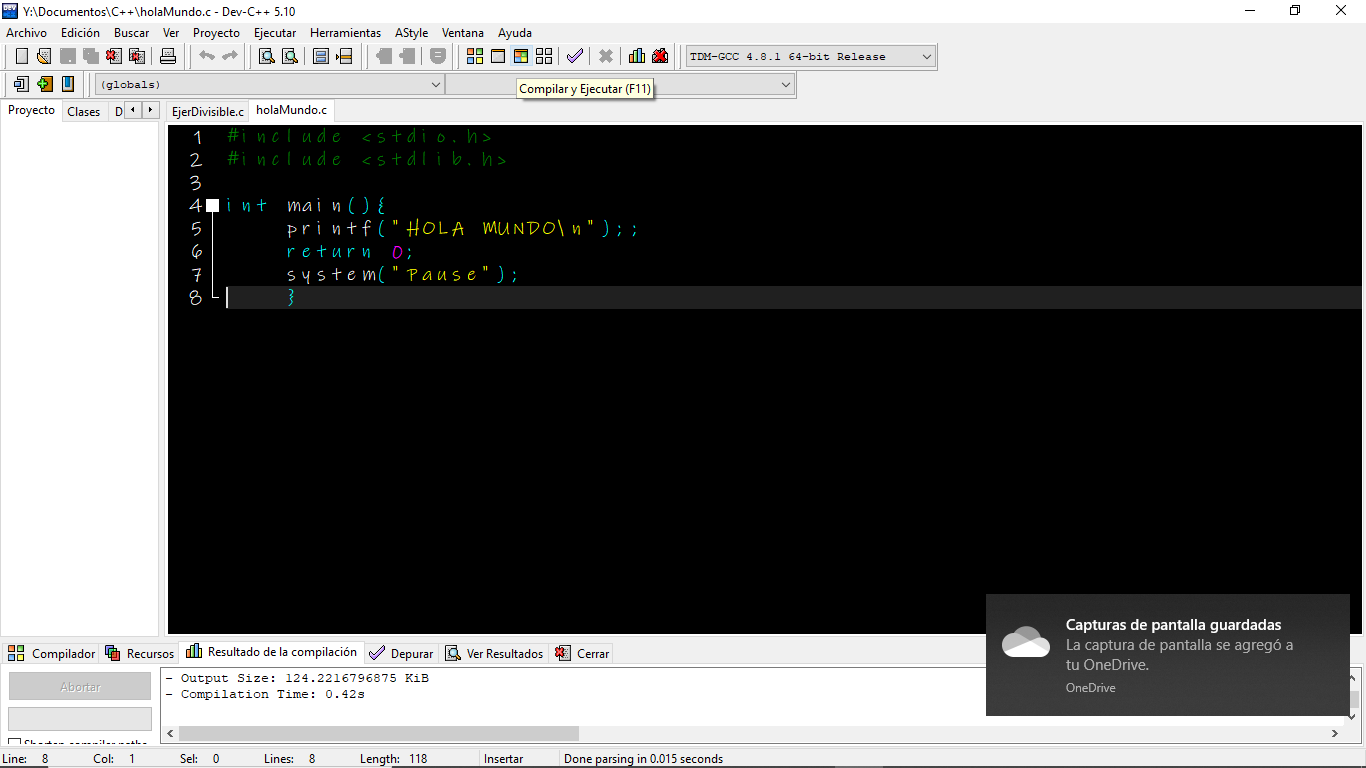
**\a** Sonido audible

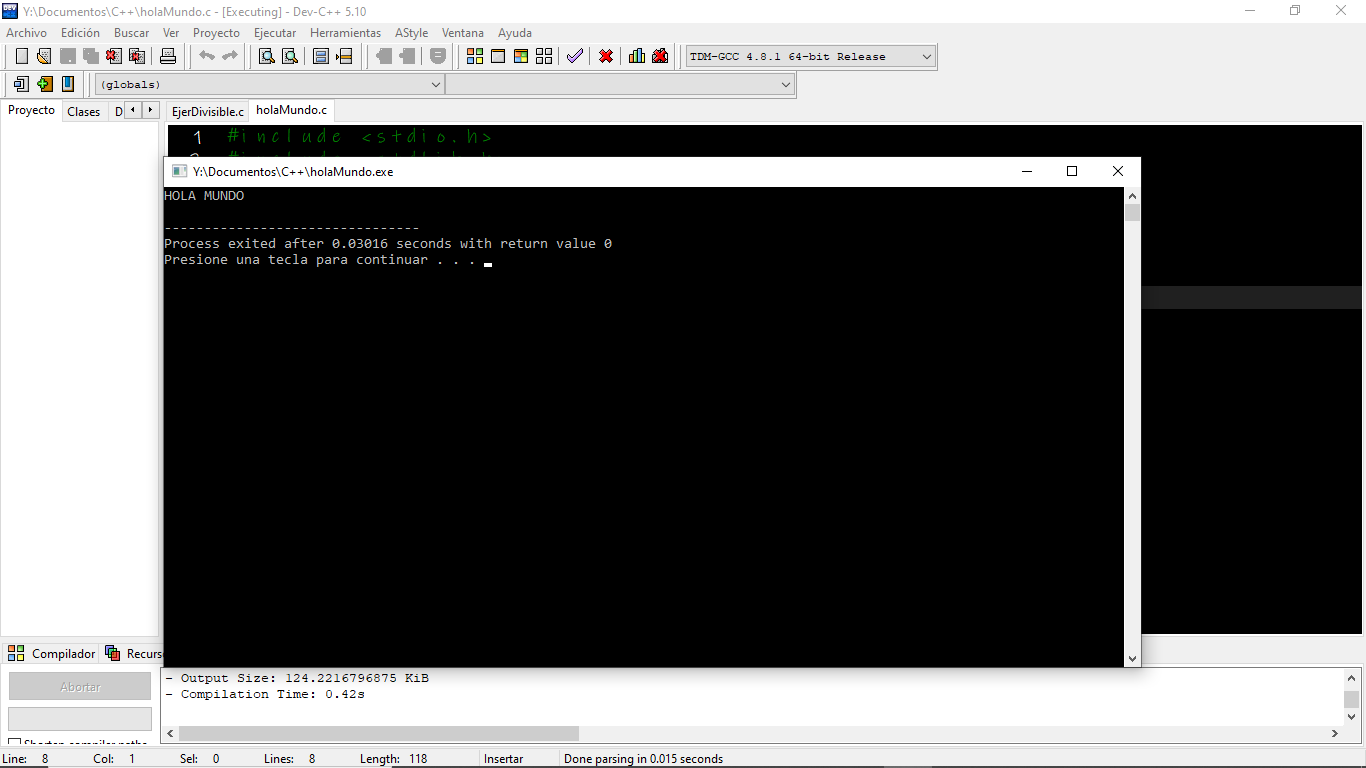
**\b** Retroceso

Siempre guardar con terminal **.c** (Hola\_Mundo.c)

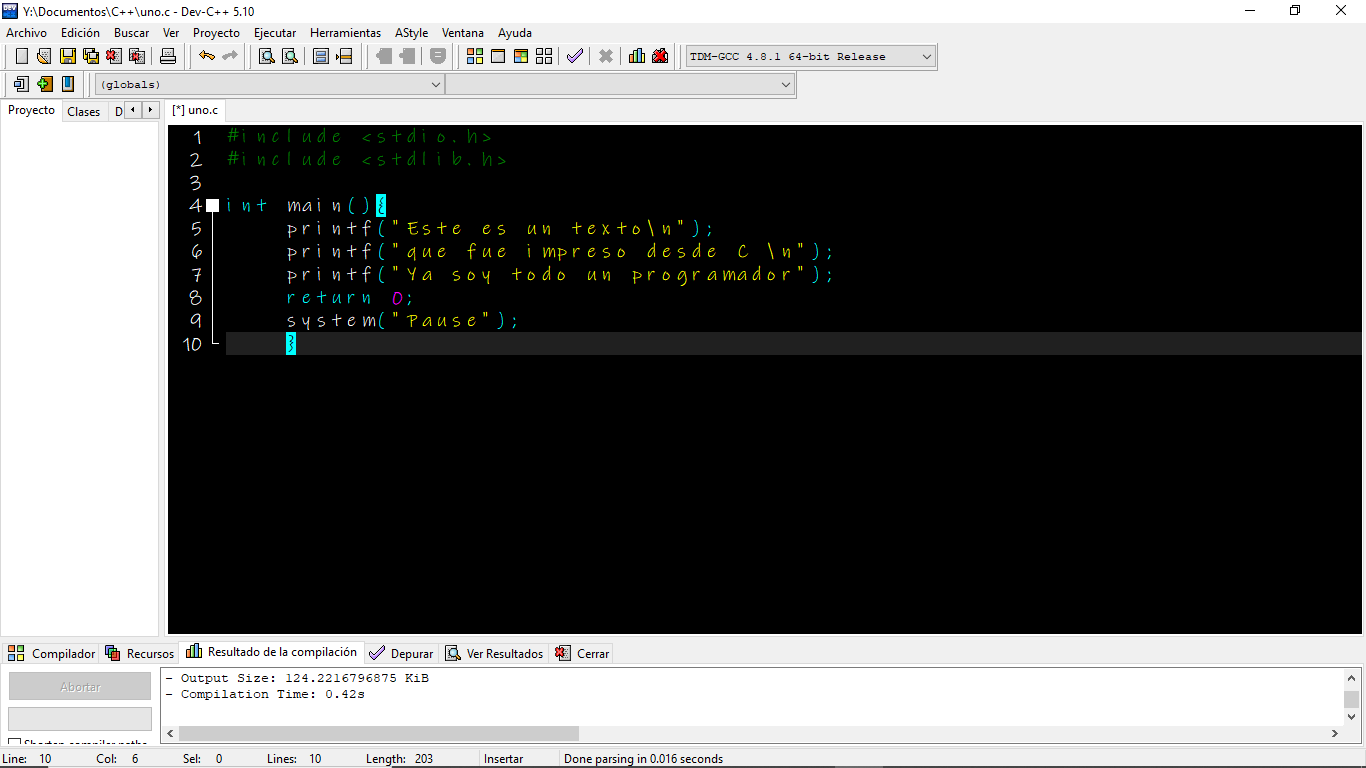


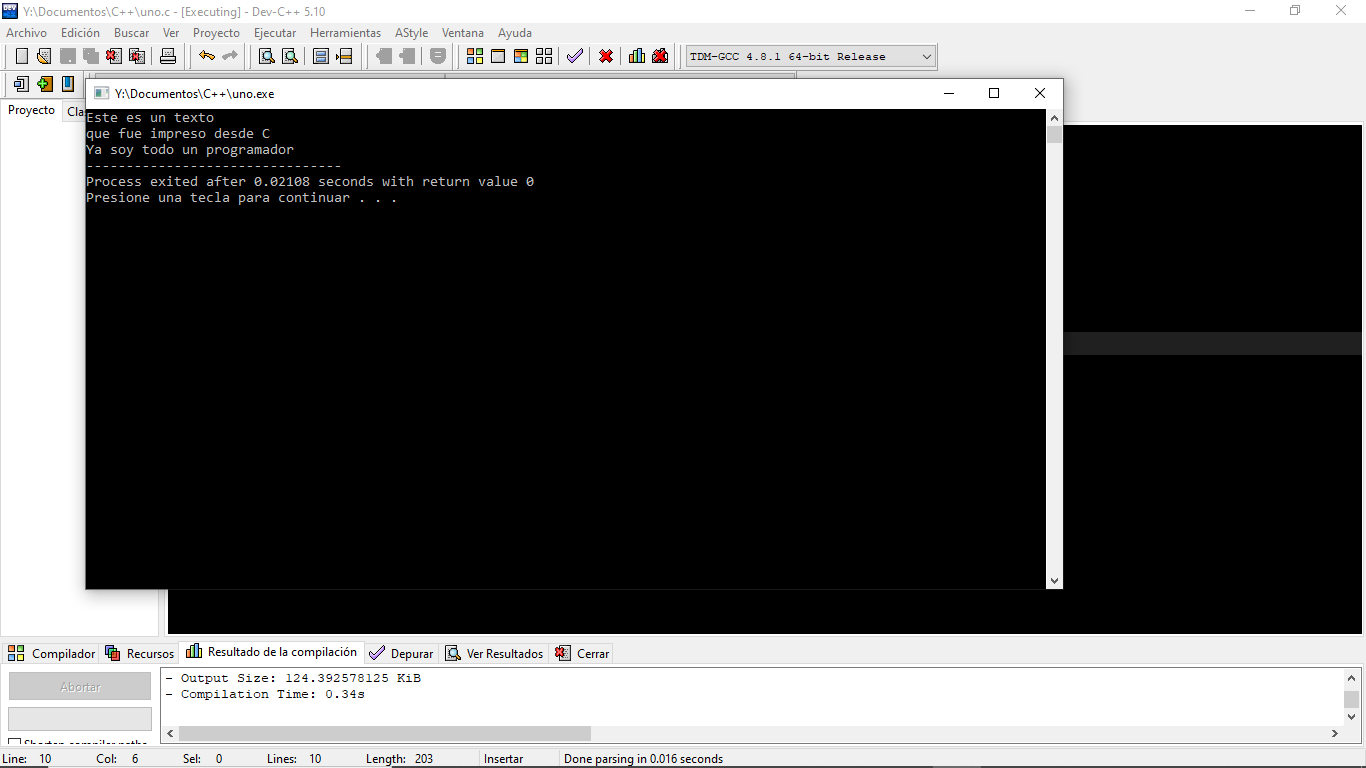
Le damos click en el botón de ejecutar y compilar.





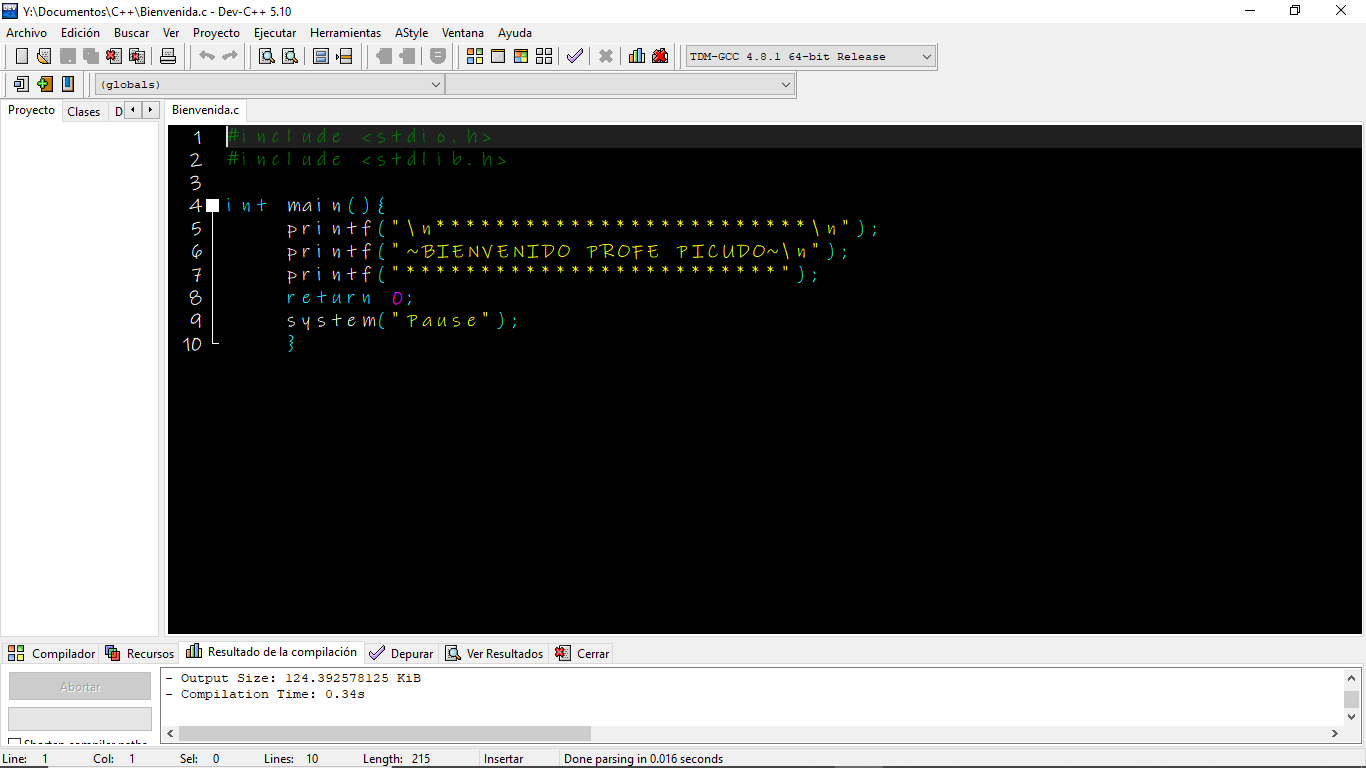
Ejemplo 2:

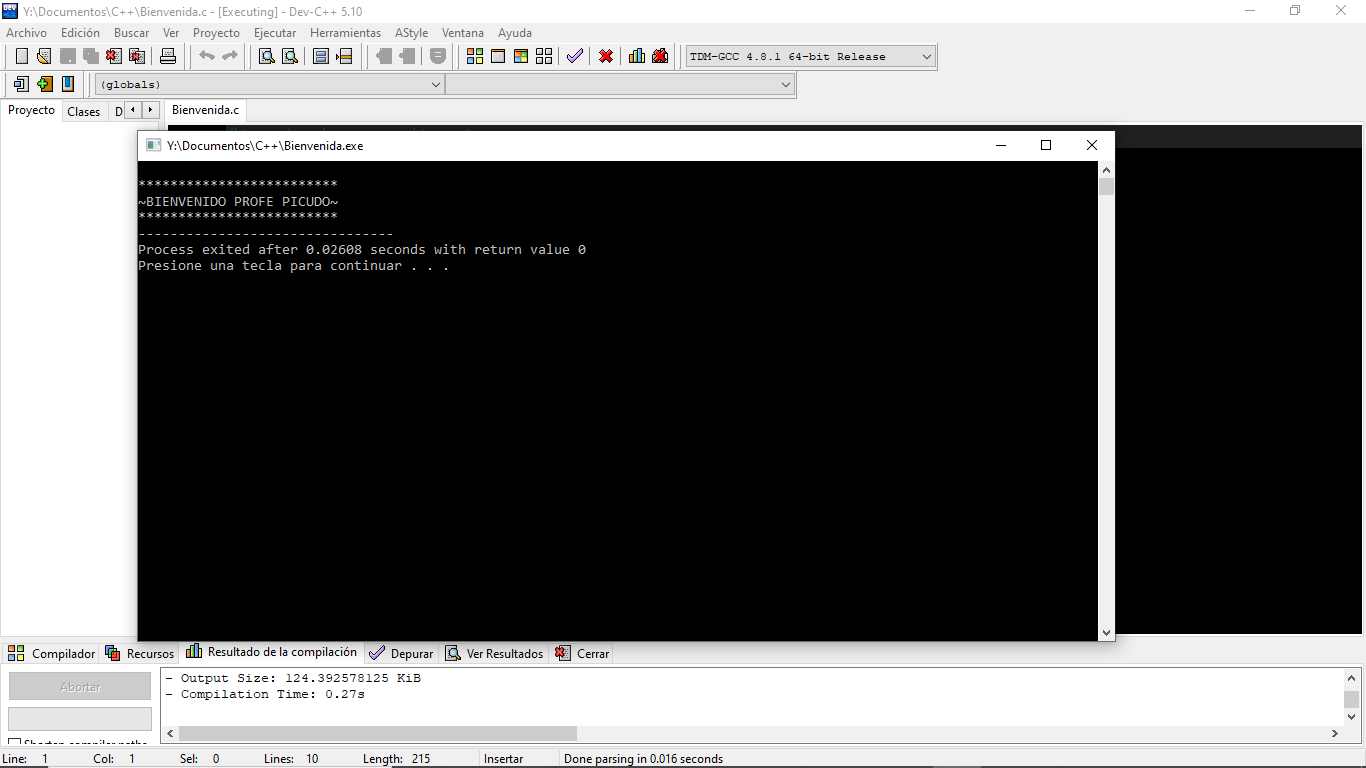




Ejercicio 3:

Crea un nuevo programa que imprima una Bienvenida con tu nombre.





Ejercicio 4:

En C hay dos tipos de comentarios:

//comentarios para una sola línea

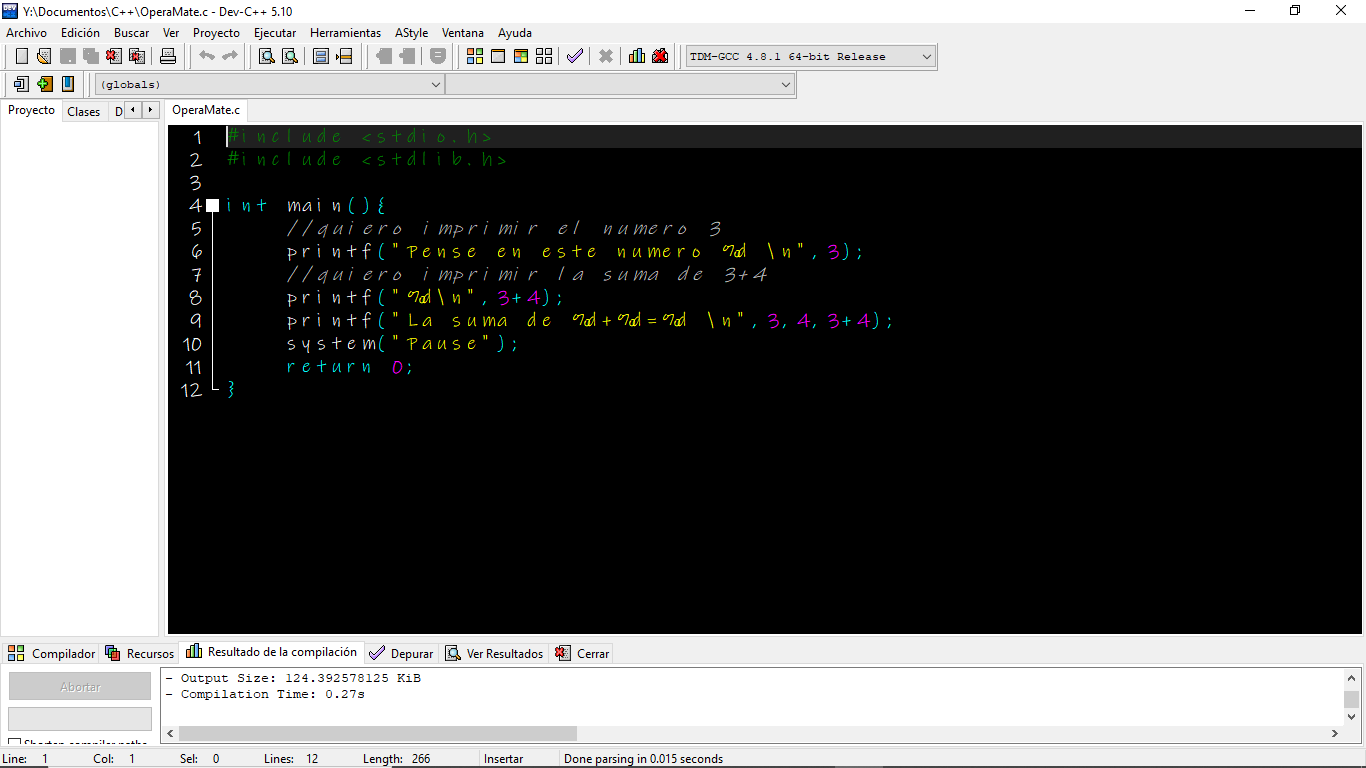
/\*

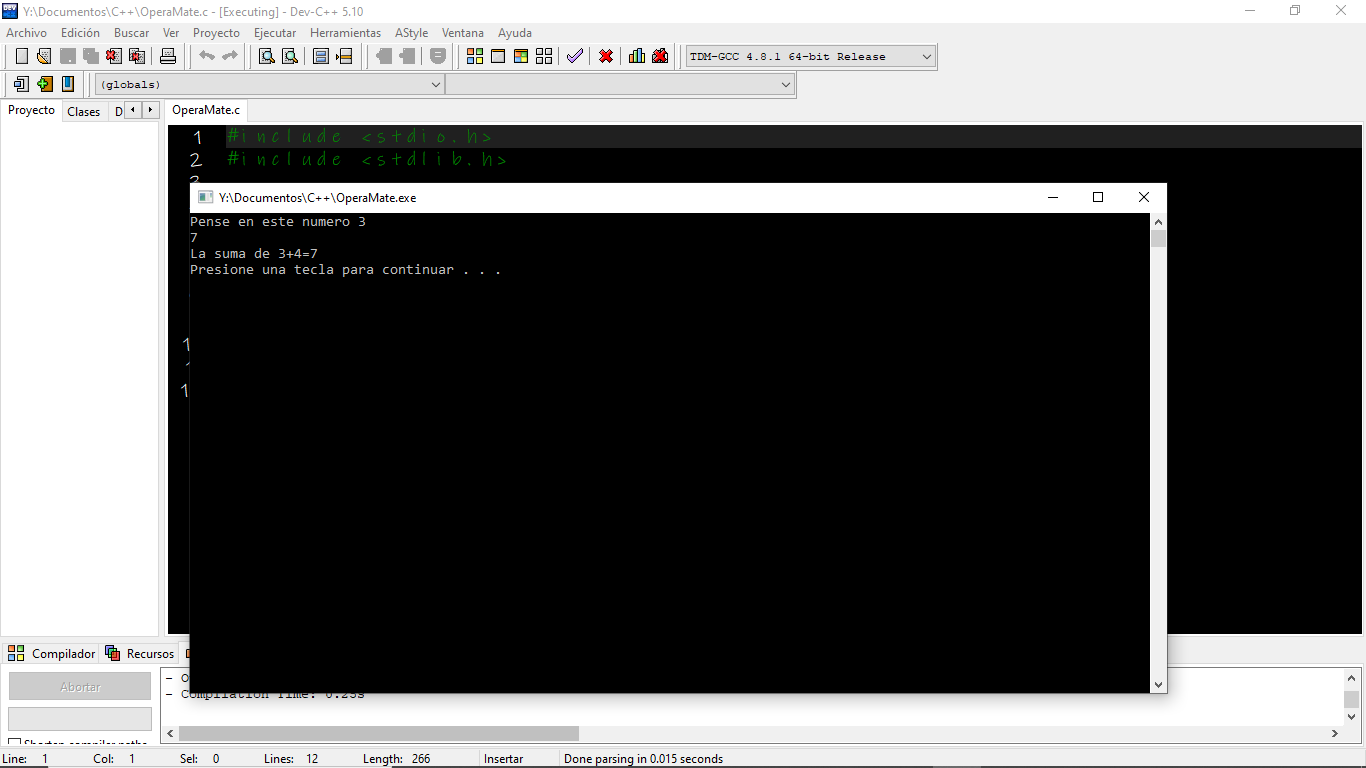
Comentarios de una o

Más líneas

\*/

%d Entero decimal de punto flotante, recibirá un tipo de dato numérico.





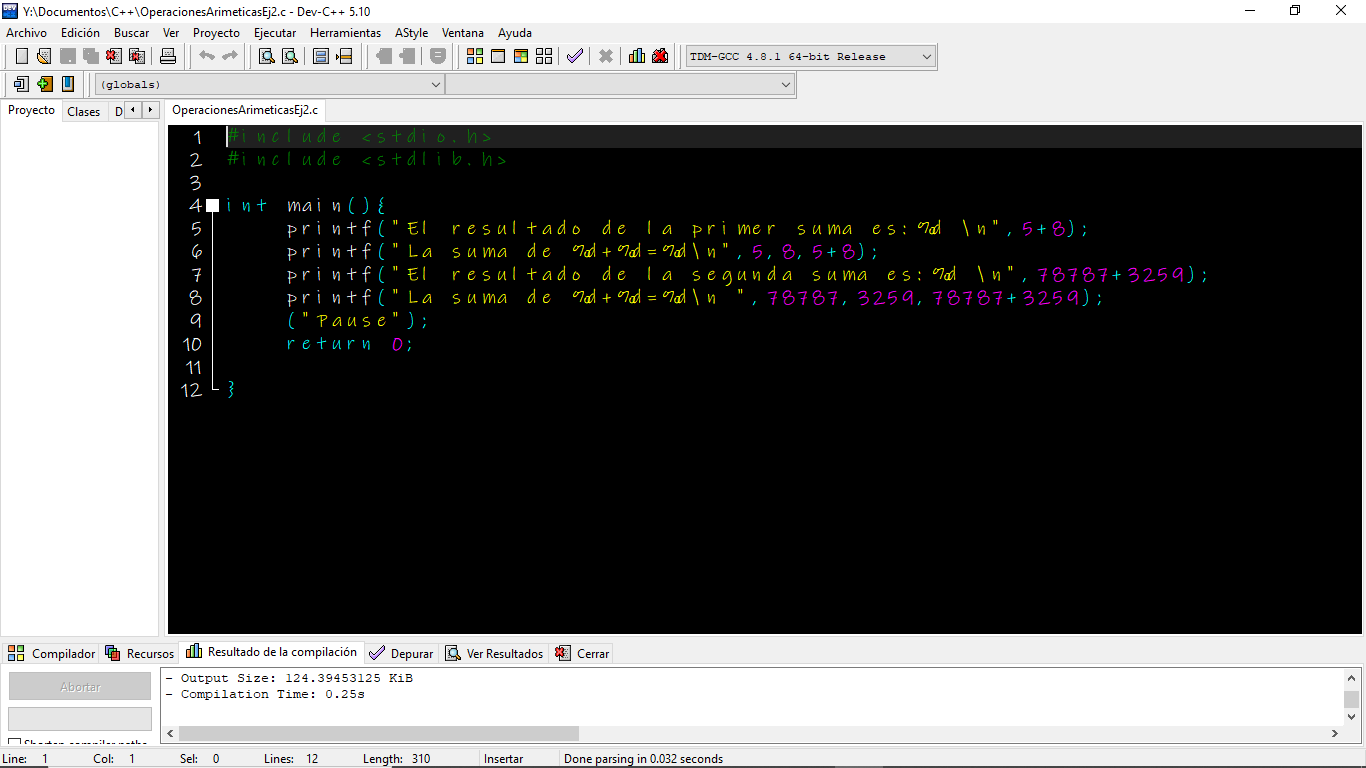
Ejercicio 5:

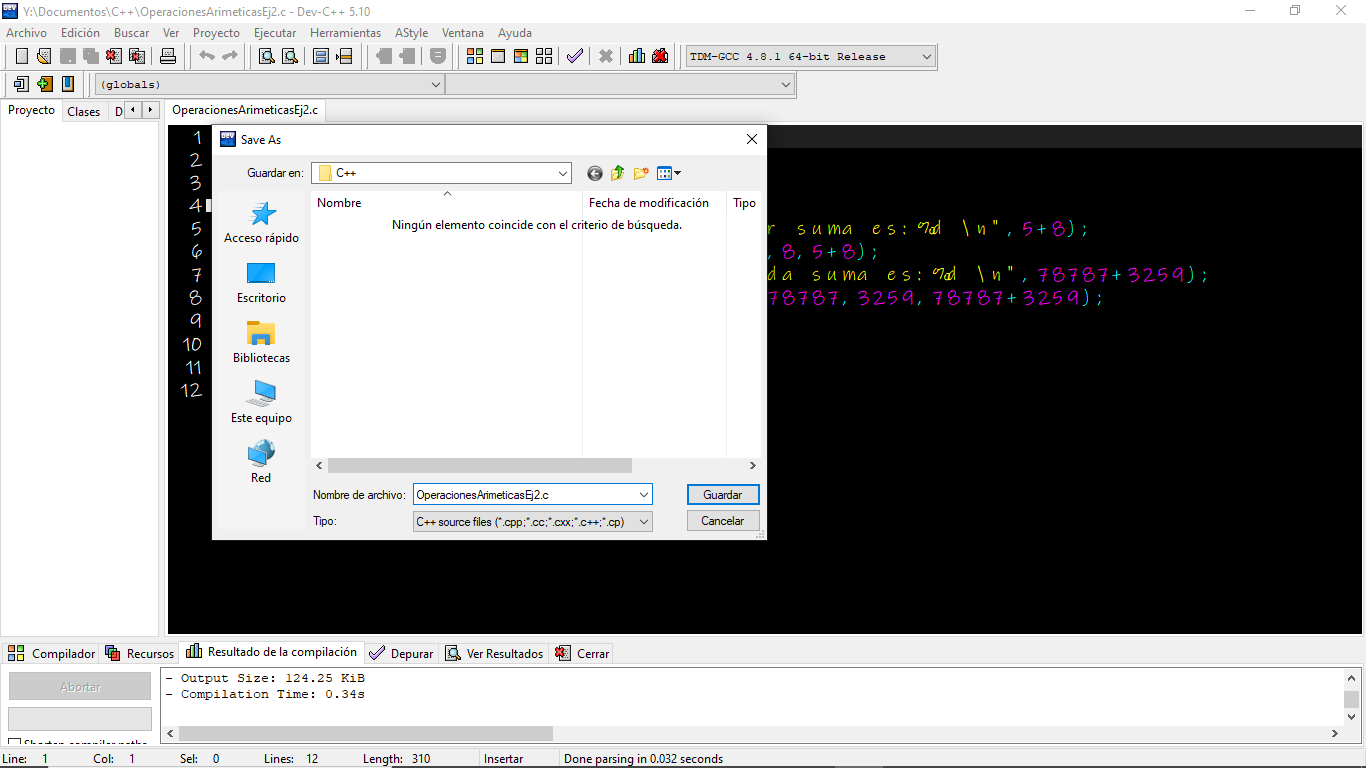
Crear un nuevo archivo en Dev-C++que se llame:

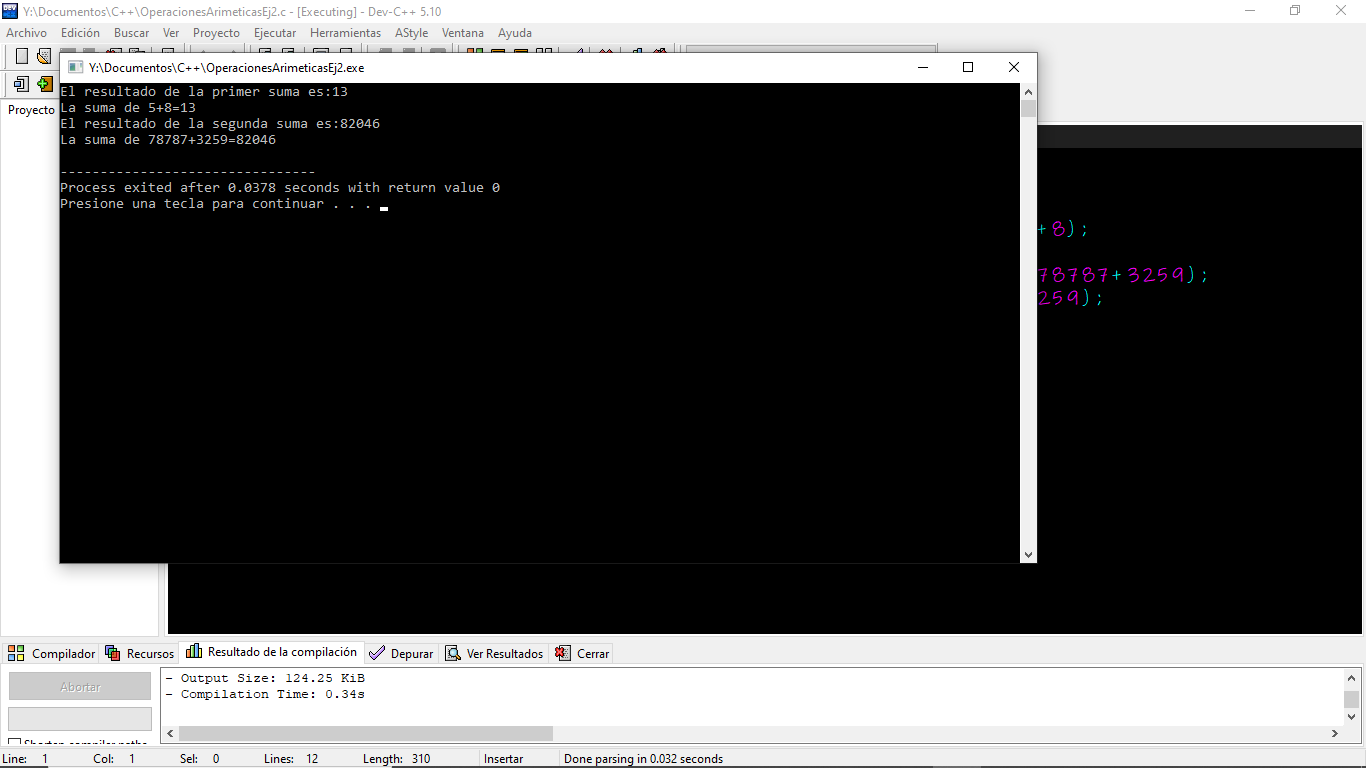
~OperacionesAritmeticas

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

\*5+8

\*78787+3259





\*OPERADORES ARIMETICOS

+ [Adición]

- [Sustracción]

\* [Multiplicación]

/ [División]

% [Módulo]

++ [Incremento]

-- [Decremento]

Palabras reservadas:

~ Las palabras reservadas son identificadores utilizados por el lenguaje para un fin especifico

~ No pueden ser empleadas por el programador.

Para nombrar variables , atributos ,métodos o clases:

**auto break case char const continue default do**

**doublé else enum extern float forgoto if int**

**long register return short signed sizeof static**

**struct typedef unión unsigned void volatile while**

Constantes:

~Una constante es un valor que no puede ser alterado durante la ejecución de un programa.

~Por conveniencia, el nombre de las constantes suele.

Escribirse en mayúsculas en la mayoría de lenguajes.

~ Las constantes de texto se escriben entre comillas dobles: "Este es un ejemplo"

~ Las constantes numéricas simplemente se escriben como son: 3.

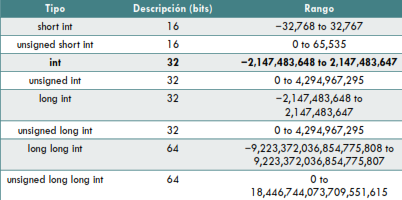
Variables

~ Localidad de memoria que permite almacenar datos

~ Los nombres de variables son una secuencia de caracteres, dígitos y guiones bajos y no deben iniciar con un dígito.

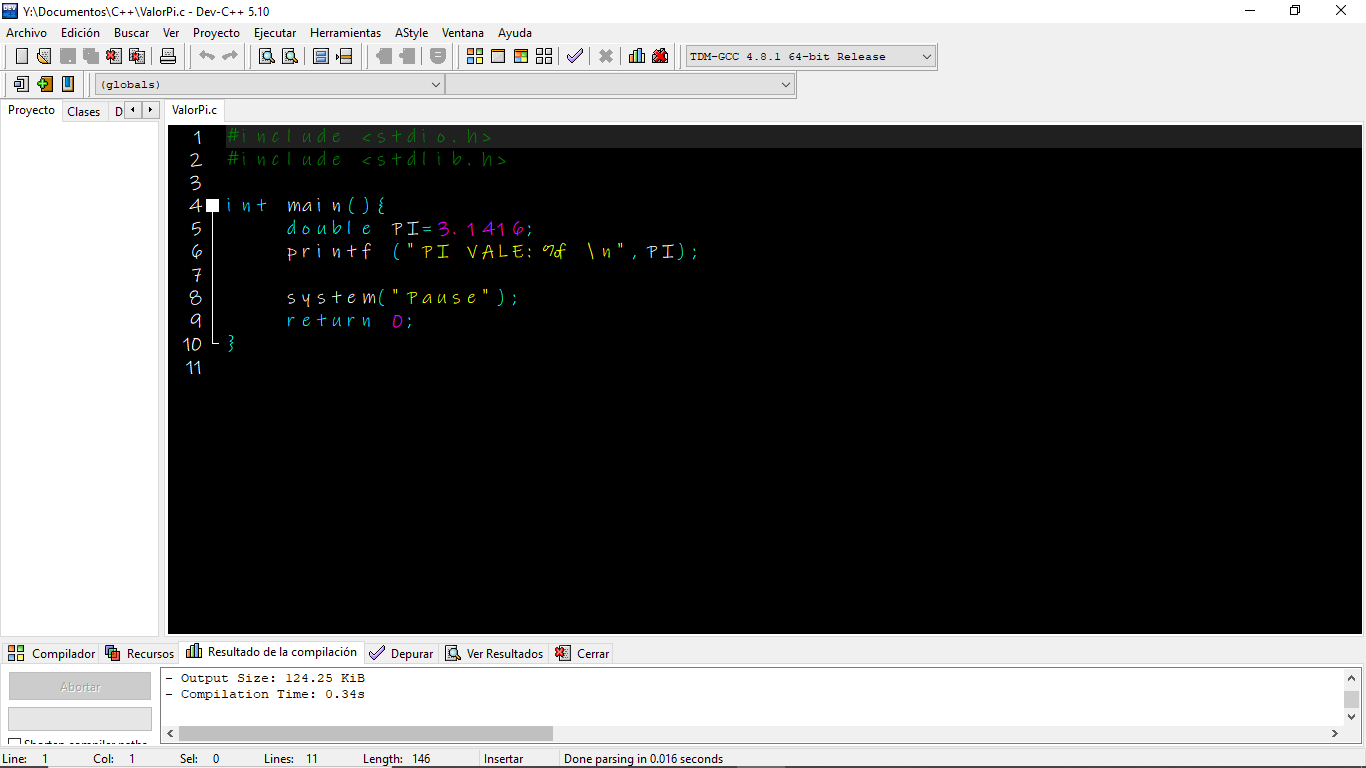
~ El valor de una variable cambia durante la ejecución del programa

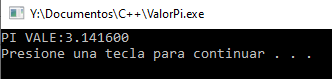




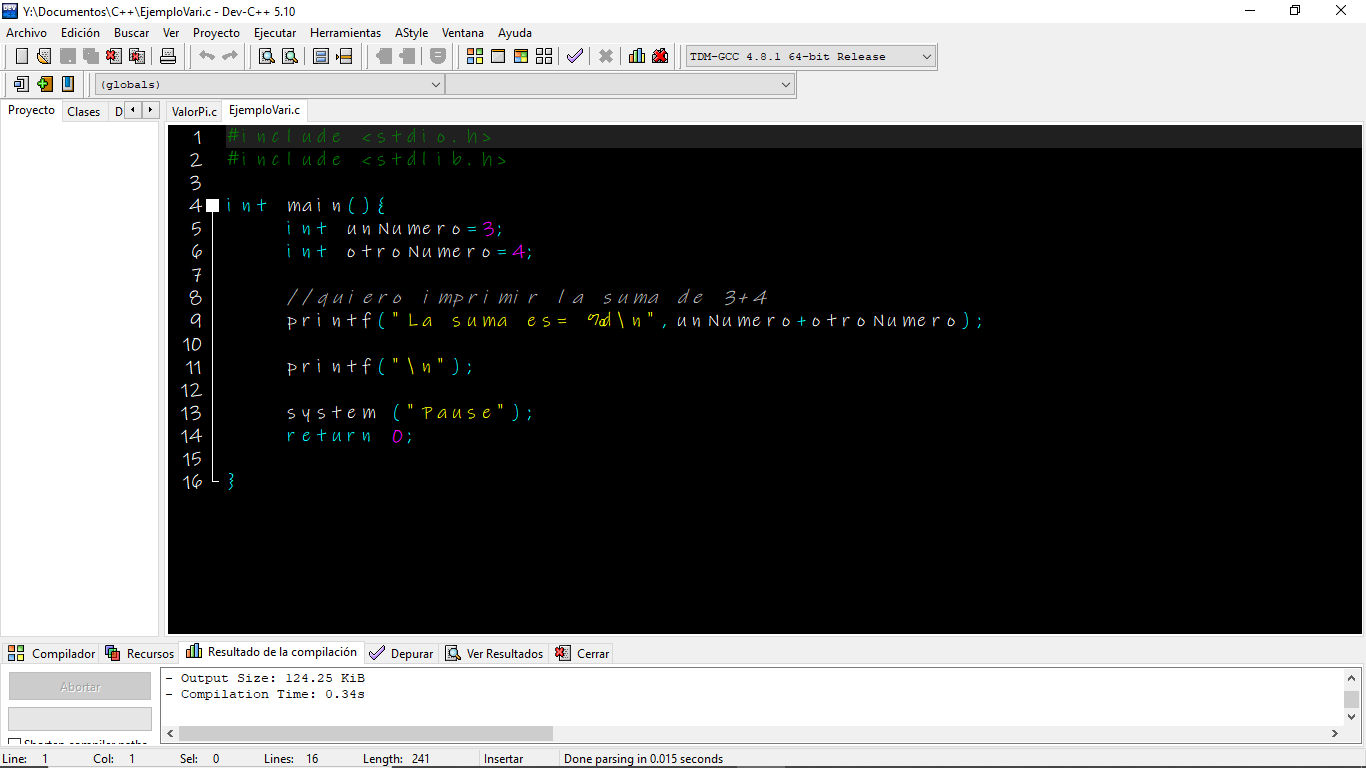


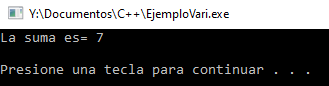
Ejemplo 6: Valor de PI





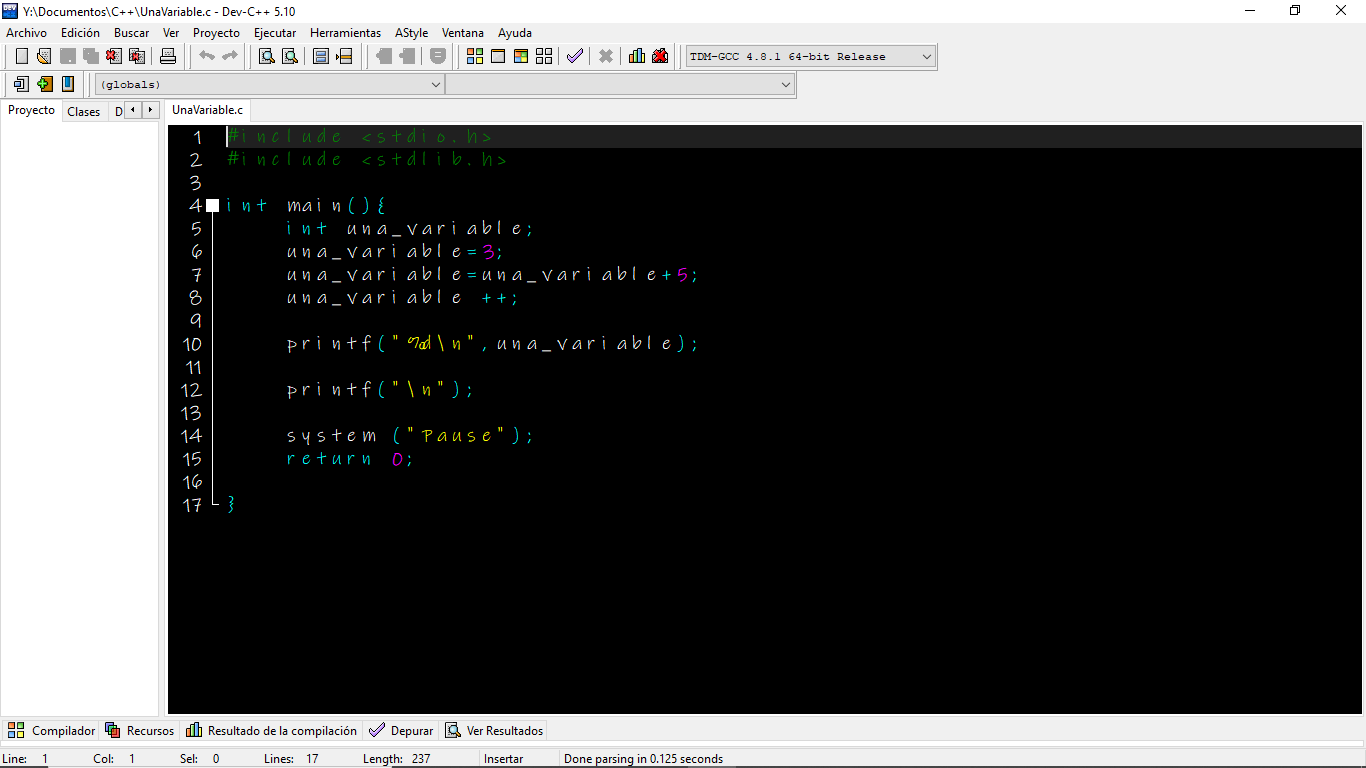
Ejemplo 7: Suma de 3+4

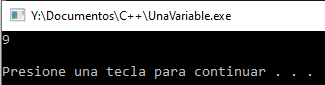




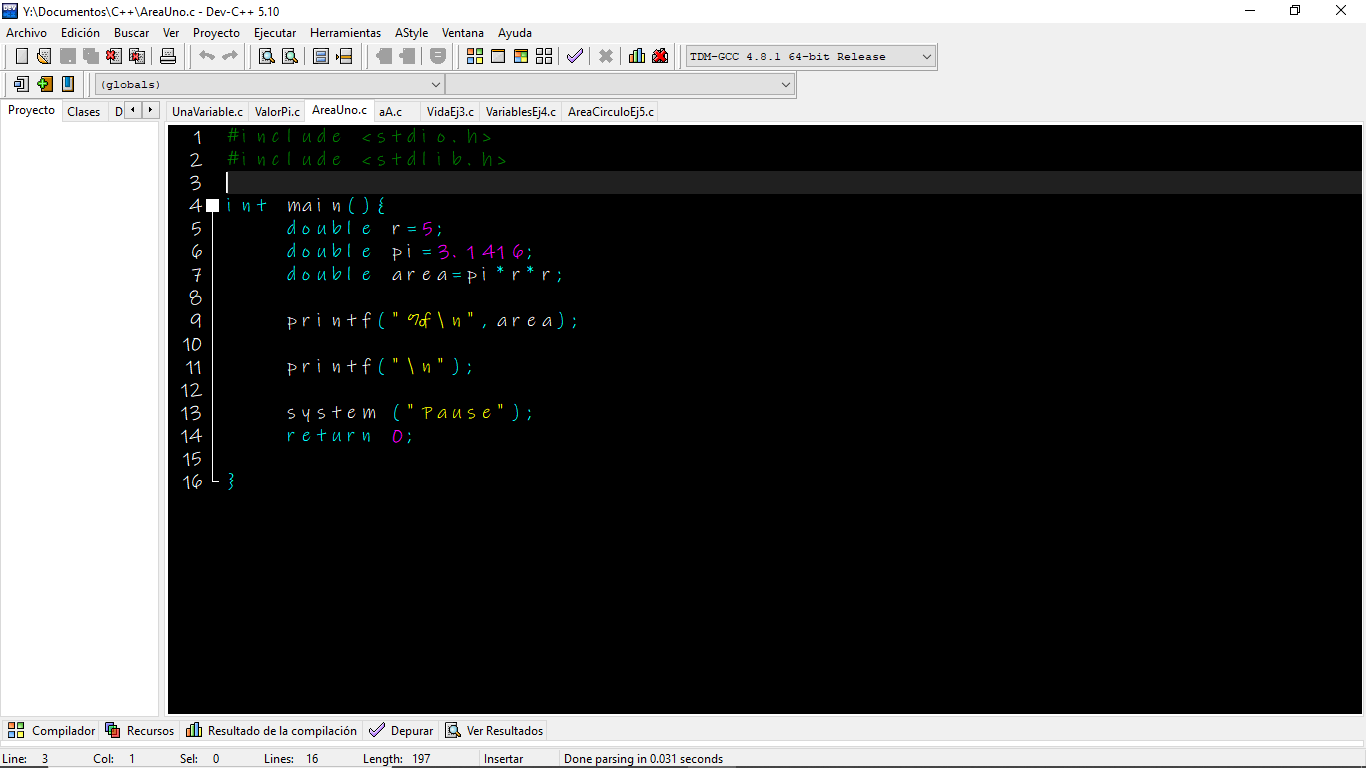
// Una variable es una referencia a una sección de memoria.

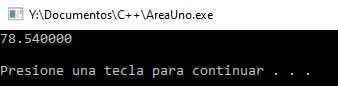
Ejemplo 8:



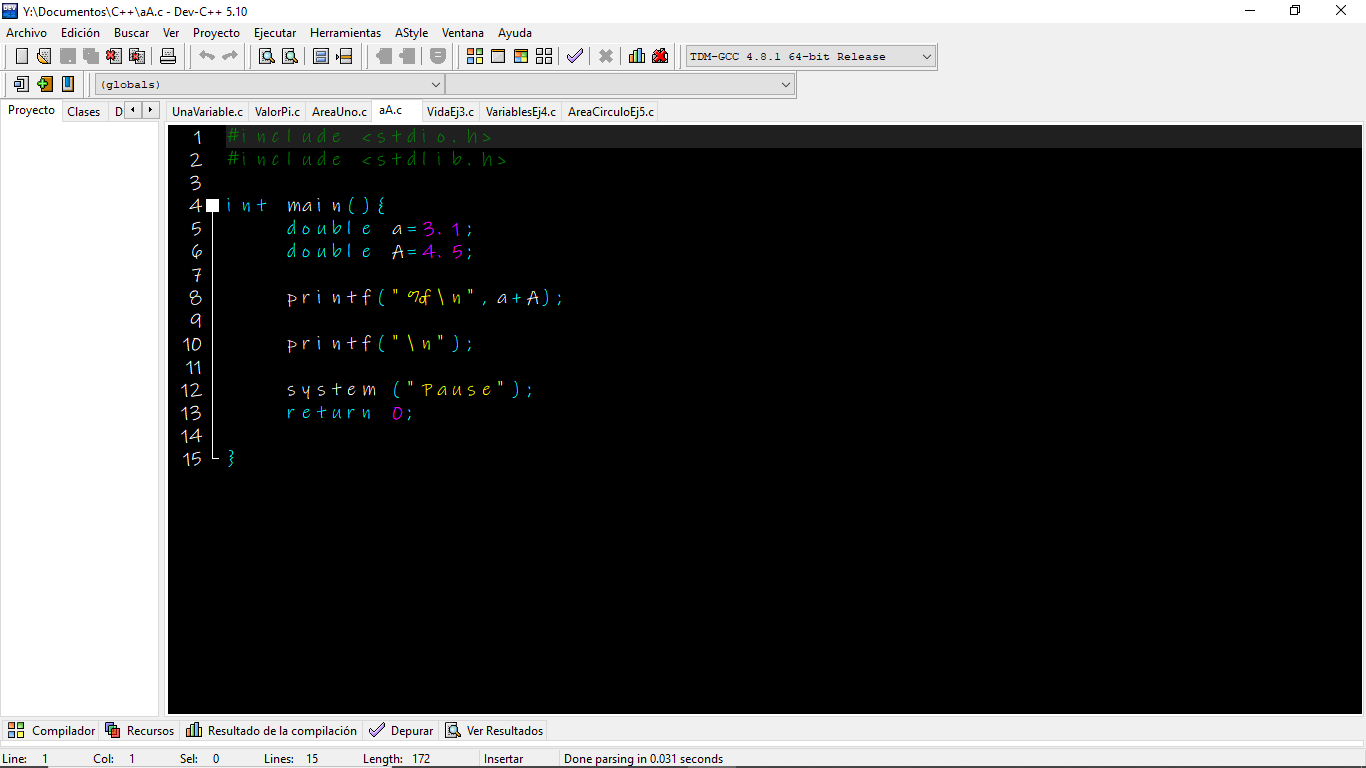


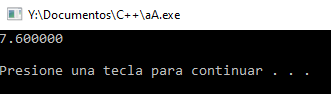
Ejemplo 9: Área del circulo





Ejemplo 10:





Ejercicio 11:

Calcula la cantidad de segundos que has vivido.

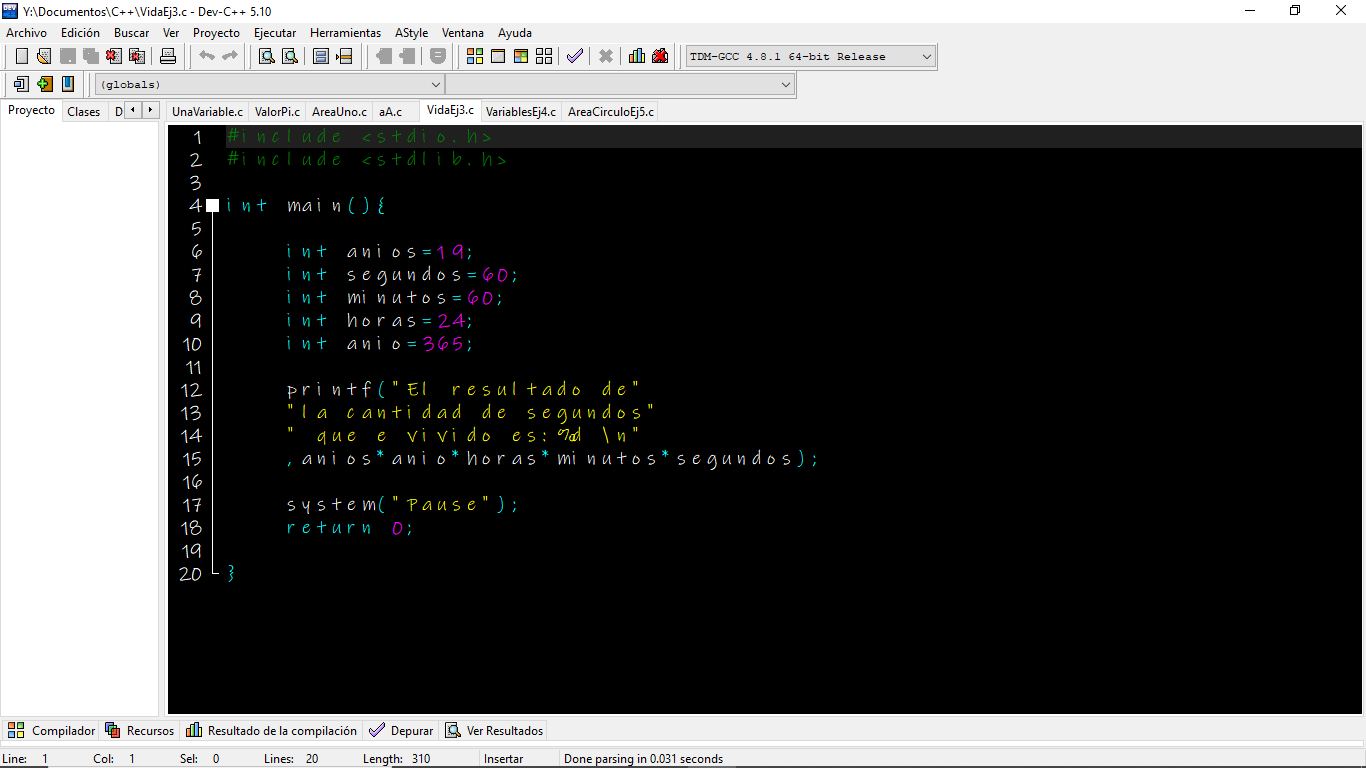
1 min = 60 seg.

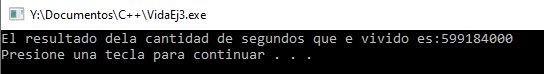
1 hora = 60 min.

1 día = 24 horas.

1 año = 365 días.

Segundos = años\*365\*24\*60\*60



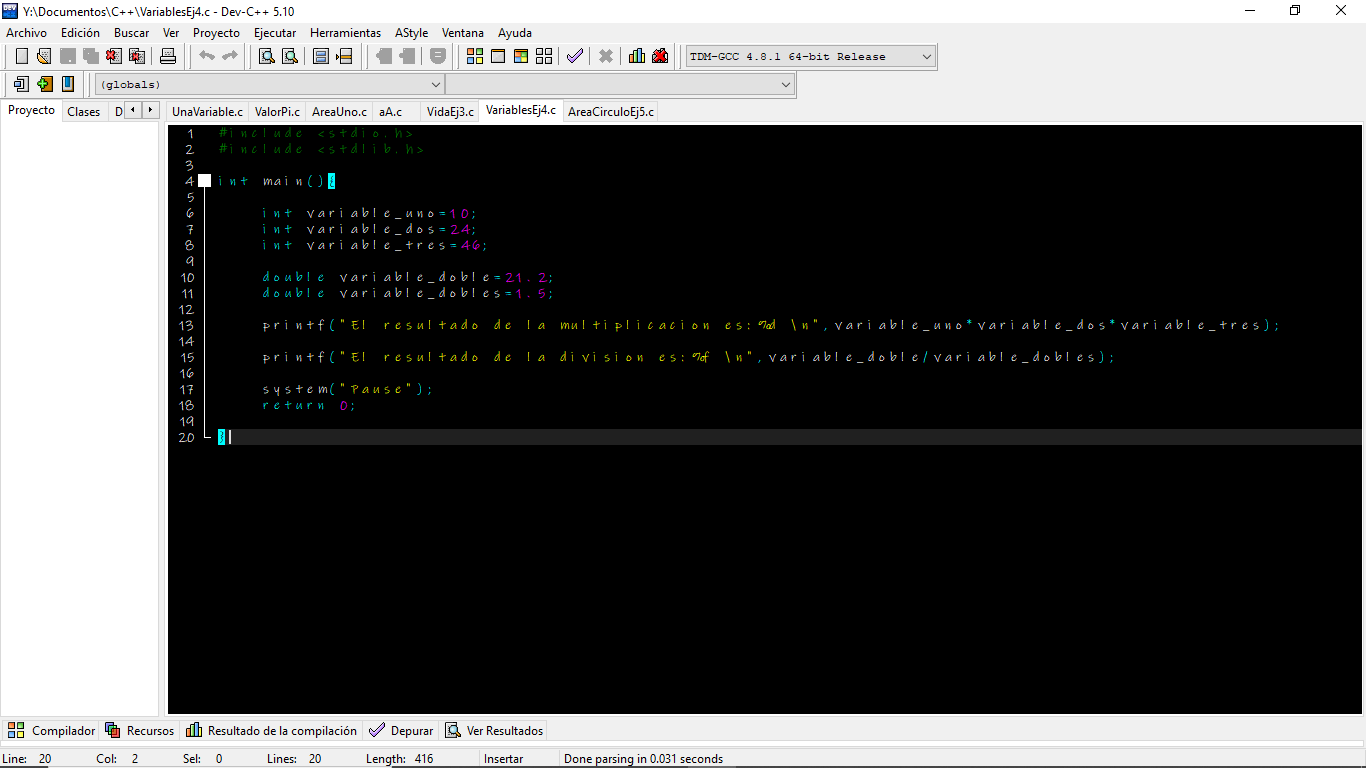


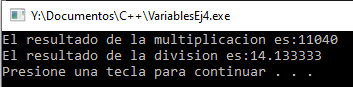
Ejercicio 12-13:

\*Declarar 3 variables y asignarles valores enteros.

\*Mostrar el resultado de la multiplicación de las 3 variables

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

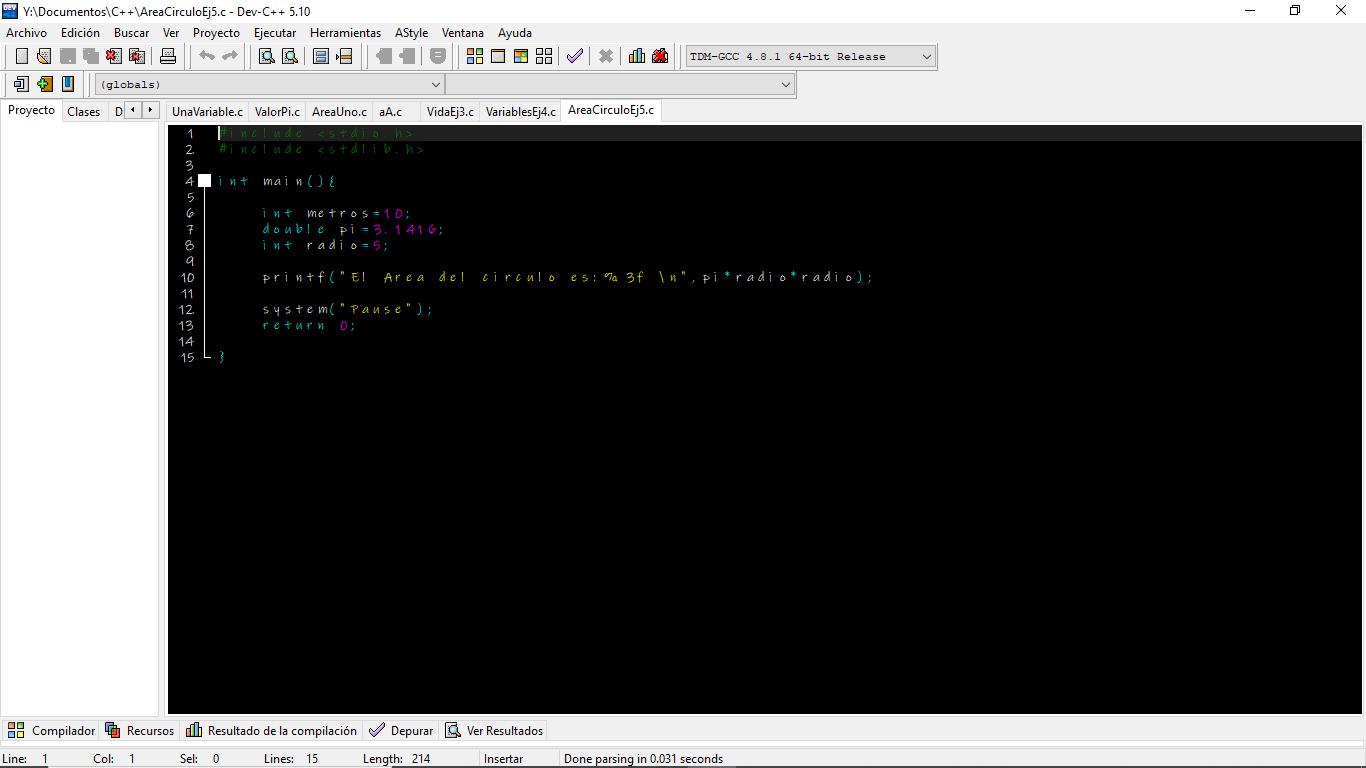
\*Mostrar el resultado de dividir la primera entre la segunda.

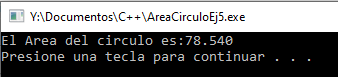


Ejercicio 14:

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

Área = pr2

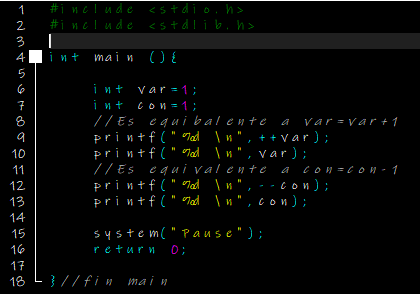


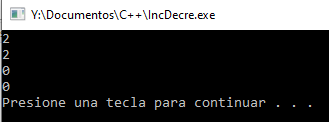


\*OPERADOR INCREMENTO/DECREMENTO

Usa el valor actual de la variable, y después le hace un incremento/decremento.

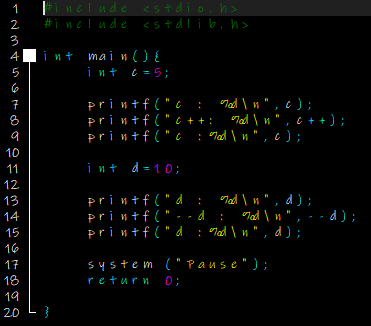
Ejemplo 15:

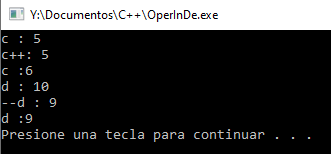


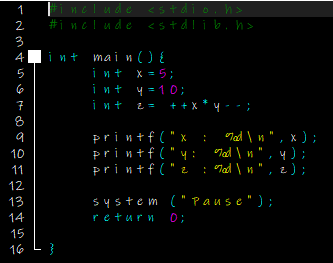


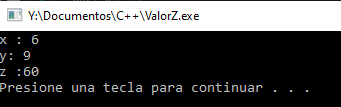
Incrementa / Decremento la variable y después la usa.

Ejemplo 16:

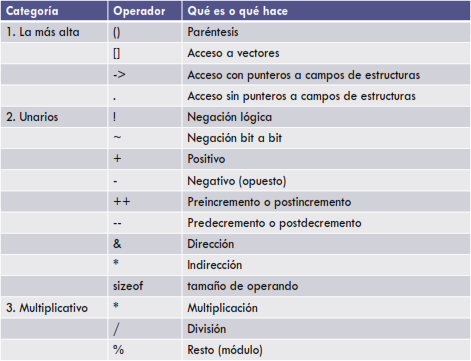


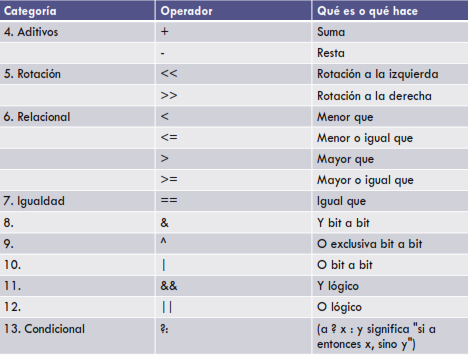


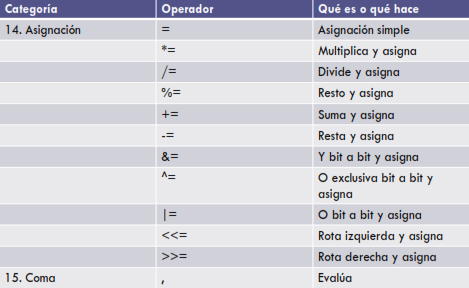
Ejemplo 17:



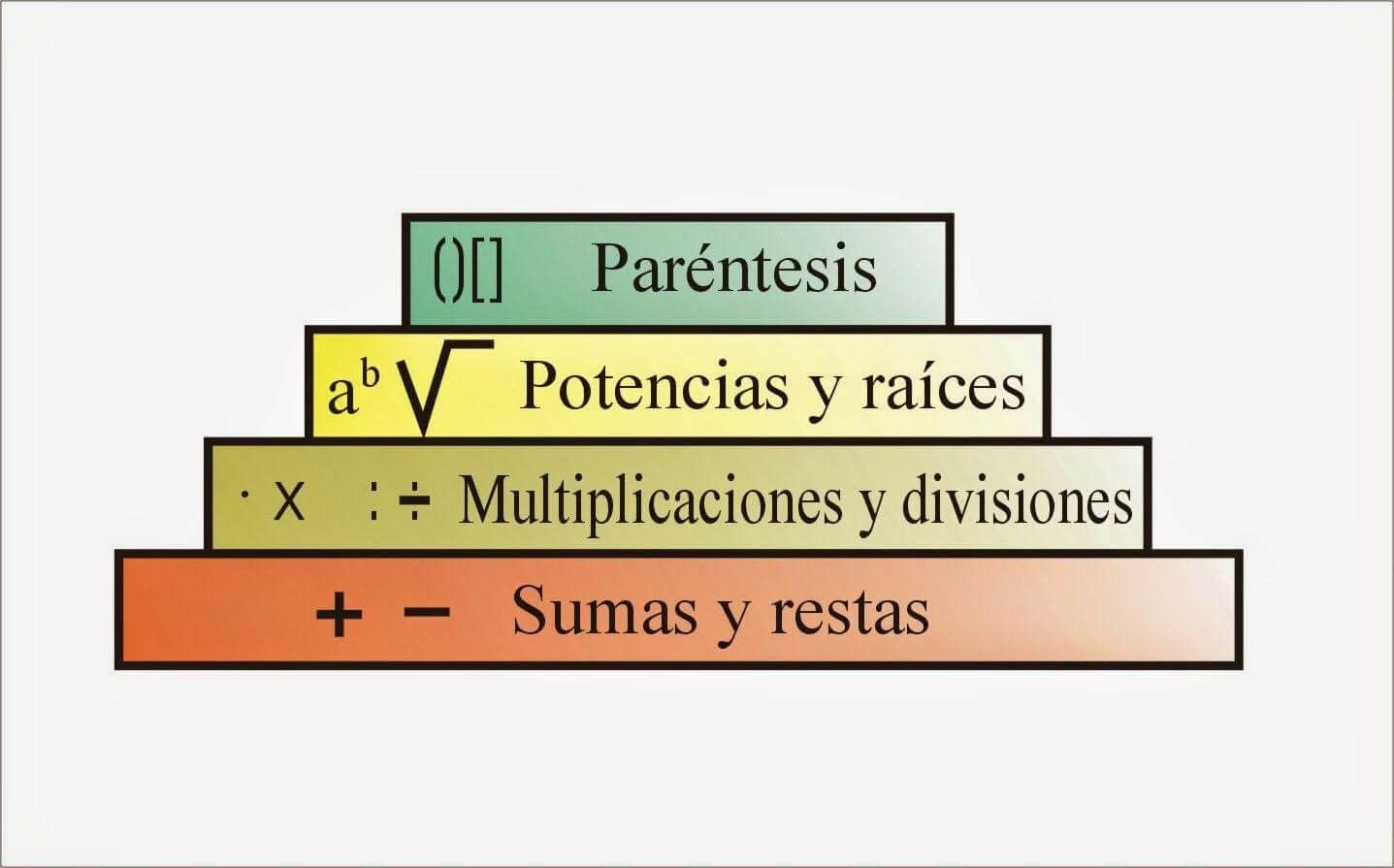
\*PRECEDENCIA DE OPERADORES

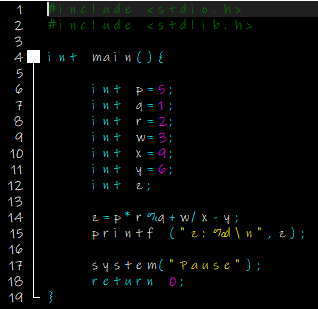


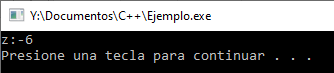


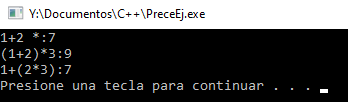


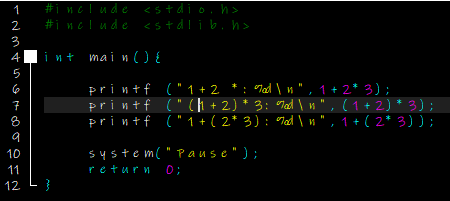
Para resolver correctamente un cálculo matemático «complejo» con varias operaciones diferentes dentro de él (sumas, divisiones, raíces cuadradas, etc.), hay que tener en cuenta el siguiente **orden de resolución:**



Ejemplo 18:

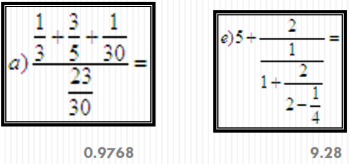




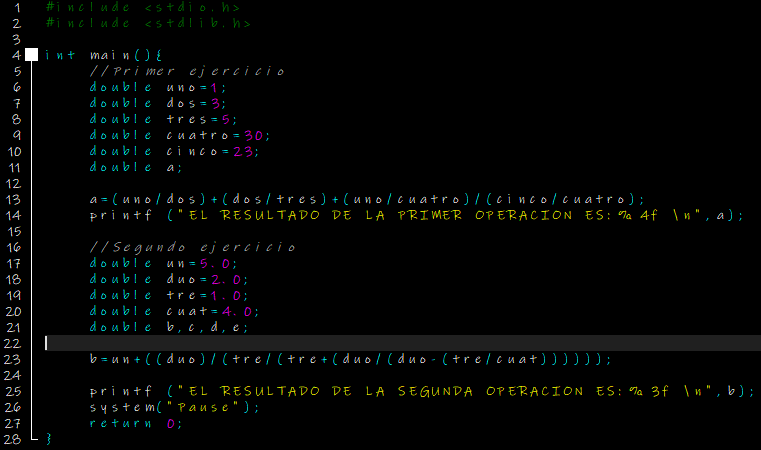
Ejemplo 19:

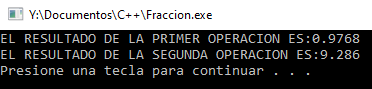
Ejercicio 20-21:

Hacer un programa en Lenguaje C que haga los siguientes cálculos y muestre los resultados en pantalla.



//El %”**.4”**f te permite solo mostrar cuantos decimales deseas imprimir en este caso colocamos **.4** para el primer ejercicio (%.4f) para que solo imprima 4 decimales y **.3** para el segundo ejercicio (%.3f) y solo imprima 3 decimales.





\*OPERADORES LOGICOS Y DE RELACION.

& Y booleano (ampersand) shift+ 6

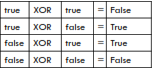
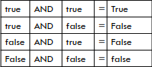
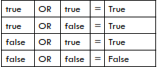
&& Y condicional (doble ampersand)

^ XOR (acento circunflejo) altgr + {

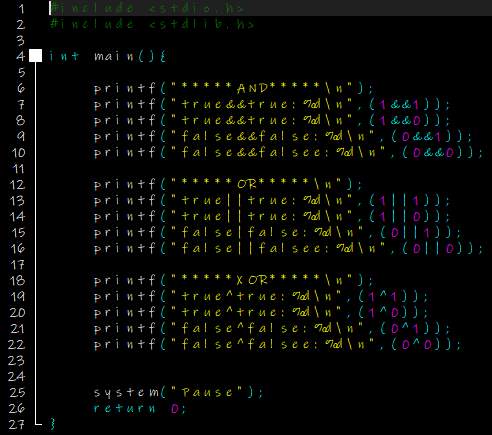
| O booleano (pipe) primer tecla numérica

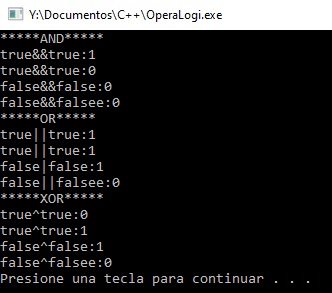
|| O condicional (doble pipe)

! NO (cierre de admiración) shift+ 1



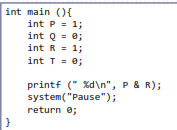
Ejemplo 22:





Ejercicio 23:

Imprimir los valores de:

~P = true

~P y R

~Q o T

~Q = false

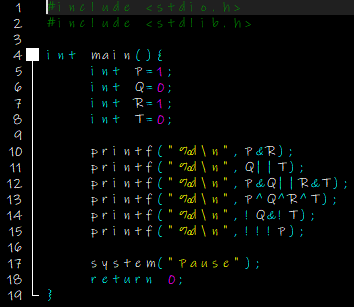
~P y Q o R y T

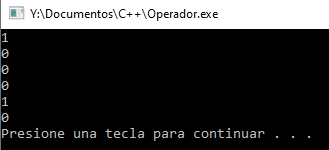
~R = true

~P xor Q xor R xor T

~not Q y not T

~T = false

~not not not P



< Menor que

<= Menor igual que

> Mayor que

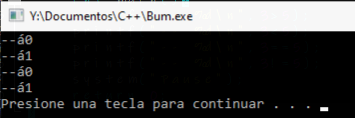
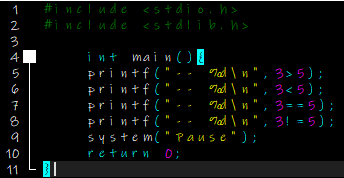
>= Mayor igual que

== Igual

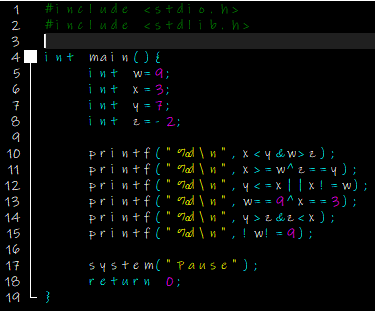
!= Diferente

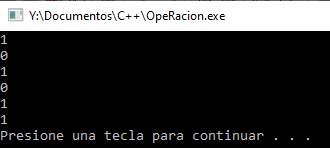
~El resultado de estos operadores es un valor booleano (true, false).

Ejemplo 24:



Ejemplo 25:





\*ENTRDADA Y SALIDA DE DATOS

SALIDA DE DATOS:

Printf()

Especifica la manera en que se formatea la salid

Imprimir un mensaje simple en pantalla

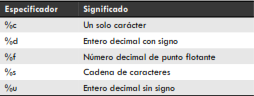


Imprimir mensajes y valores de variables



Se debe incluir el archivo de encabezado STDIO.H

ESPESIFICADOR DE CONVERSION



Puts()

Se usa para imprimir mensajes en pantalla, pero no puede desplegar variables numéricas.

Se pueden incluir secuencias de escape en una cadena.

Se debe incluir el archivo de encabezado STDIO.H



ENTRADA DE DATOS:

Scanf()

**10** 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.

Usa el espacio en blanco para separar la entrada en campos.