|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre de la práctica** | **Lenguaje C** | | | **No.** | **1** |
| **Asignatura:** | **Métodos numéricos** | **Carrera:** | **INGENIERIA EN SISTEMAS COMPUTACIONALES** | **Duración de la práctica (Hrs)** |  |

**I. Competencia(s) específica(s):** \* \* \* \* \* \* \* \* \*

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

* Aula

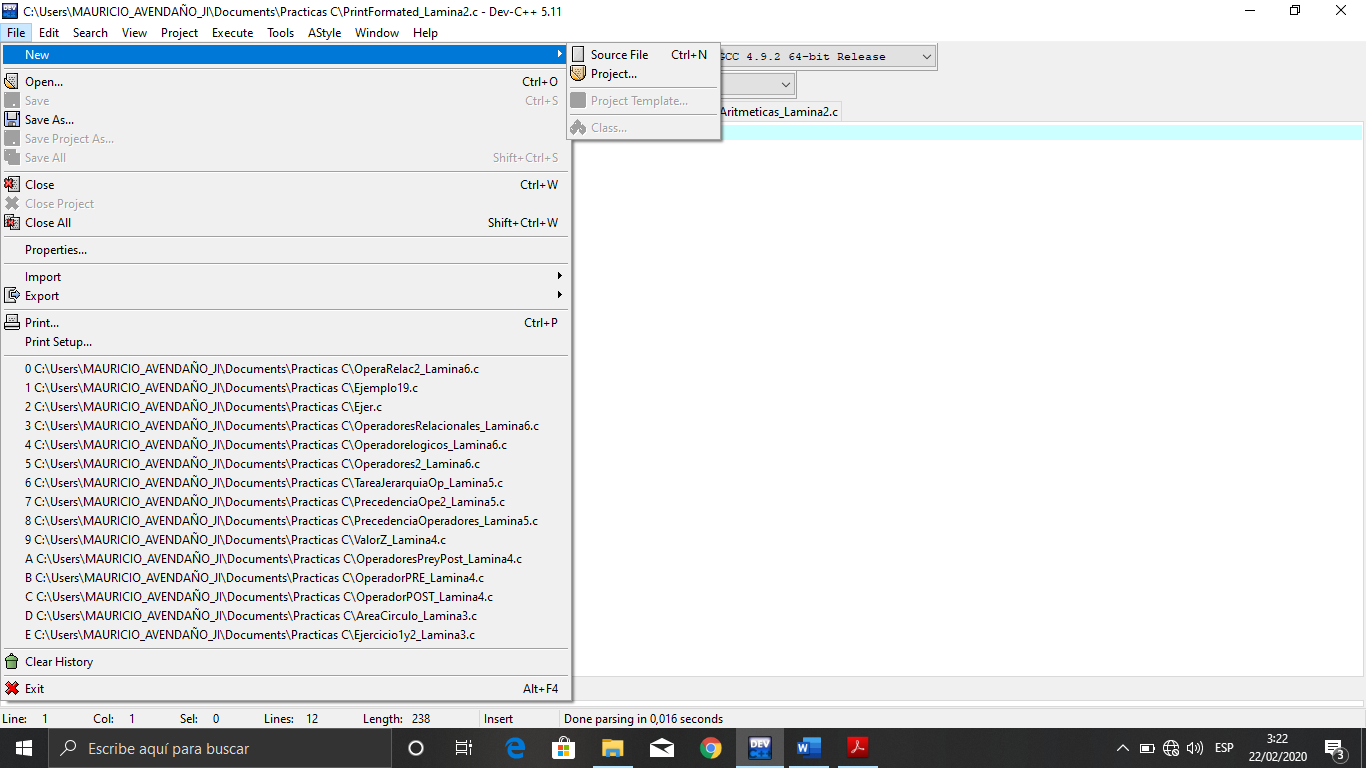
**III. Material empleado:**

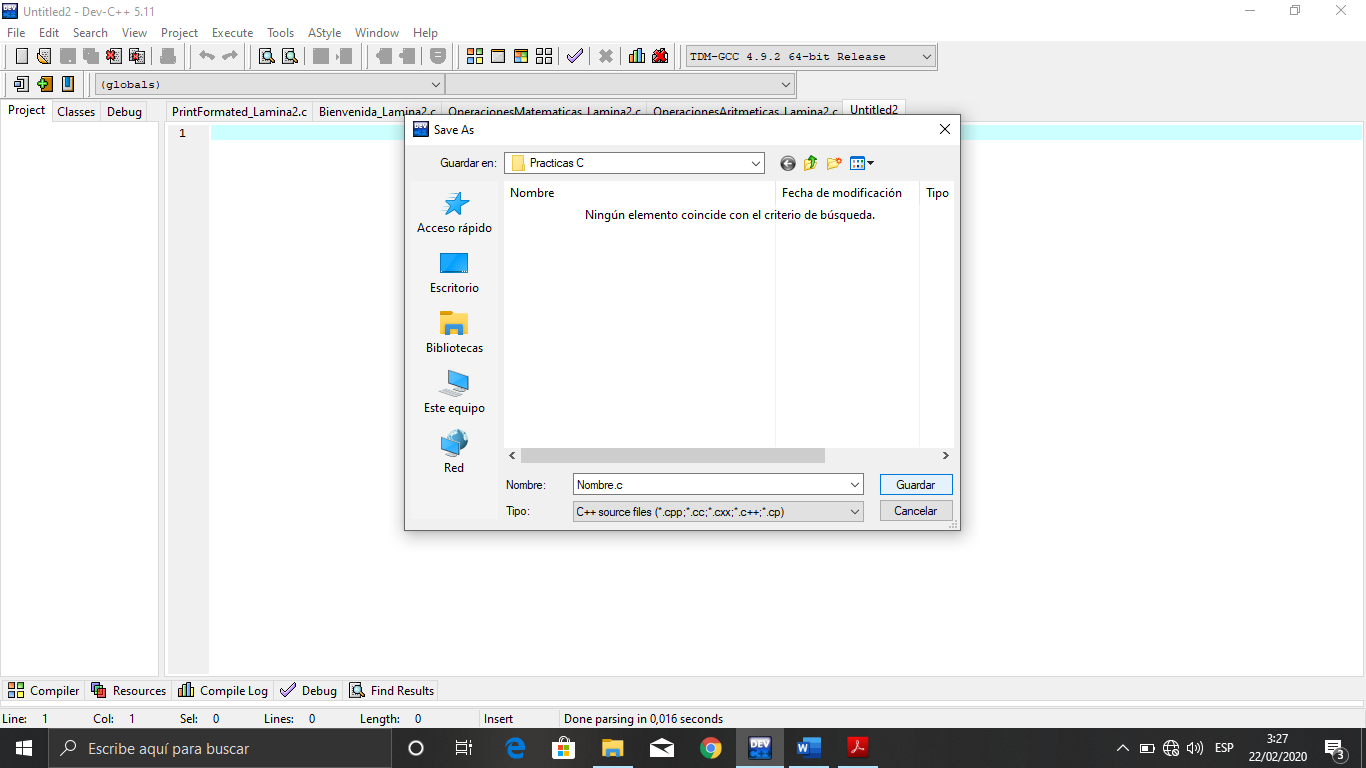
* Dev-C++
* Editor de texto
* Documentos

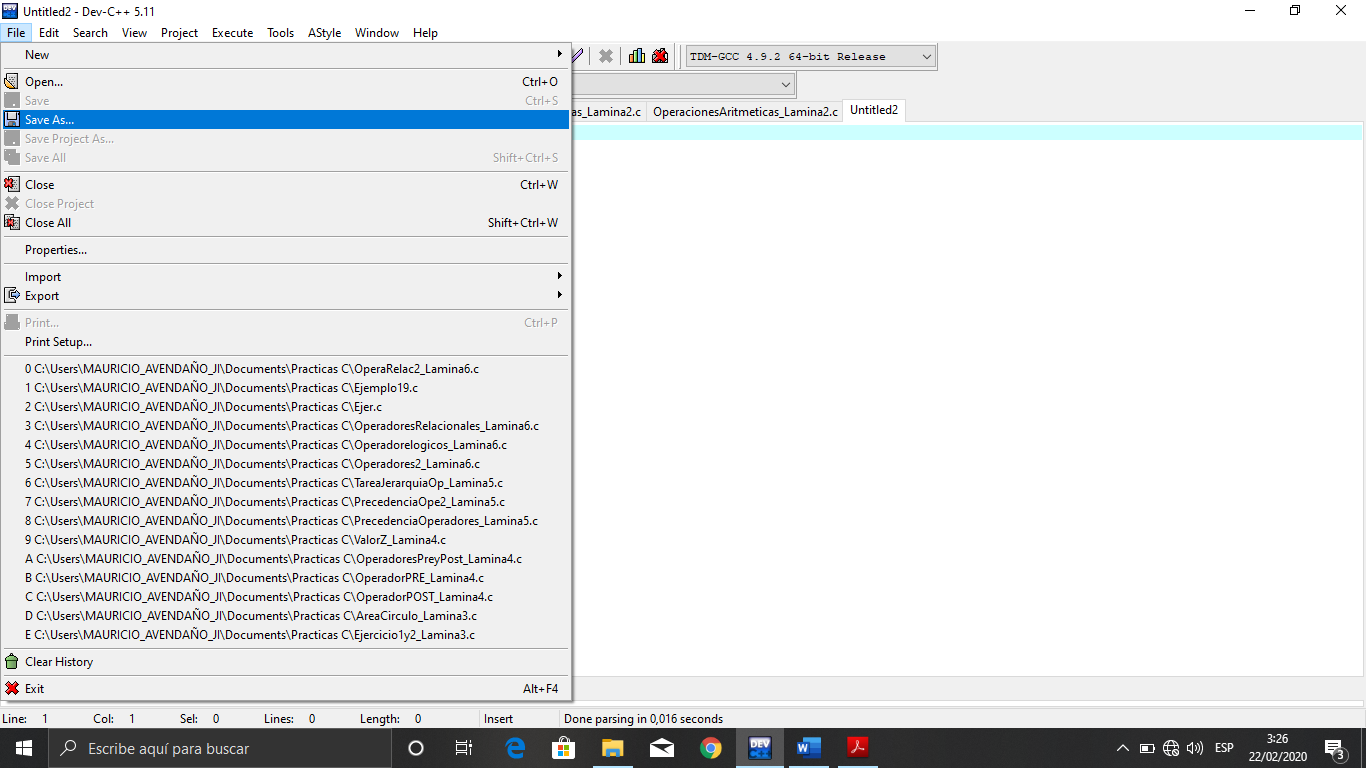
**IV. Desarrollo de la práctica:**

Para crear un archivo programado en C:

* Ejecutar el software
* En la parte superior izquierda seleccionar File>New>Source File

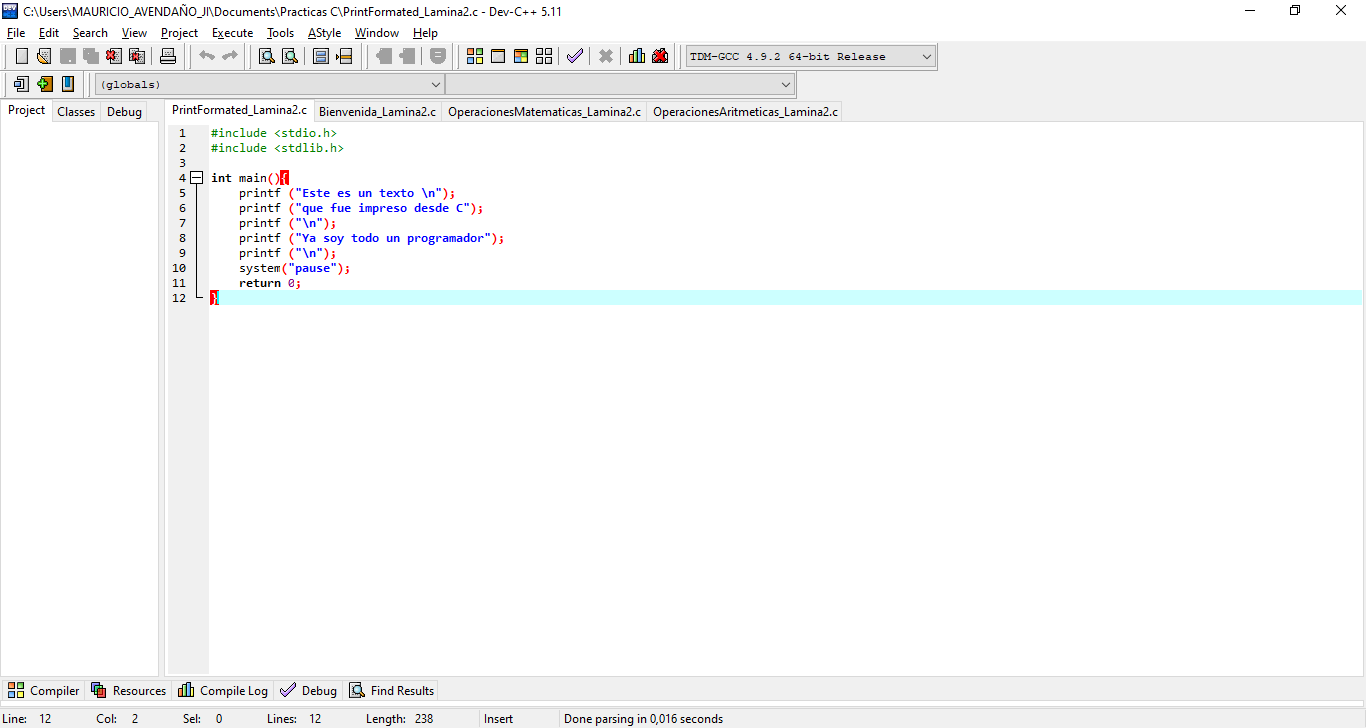


* Una vez que se haya creado, nombrar al archivo y guardarlo con la extensión C. para guardarlo ir a la pestaña File> Save as



**Documento 2**

Ejemplo. Printf (print formated).



Función que nos permite detener la ejecución del programa y visualizar los resultados en pantalla

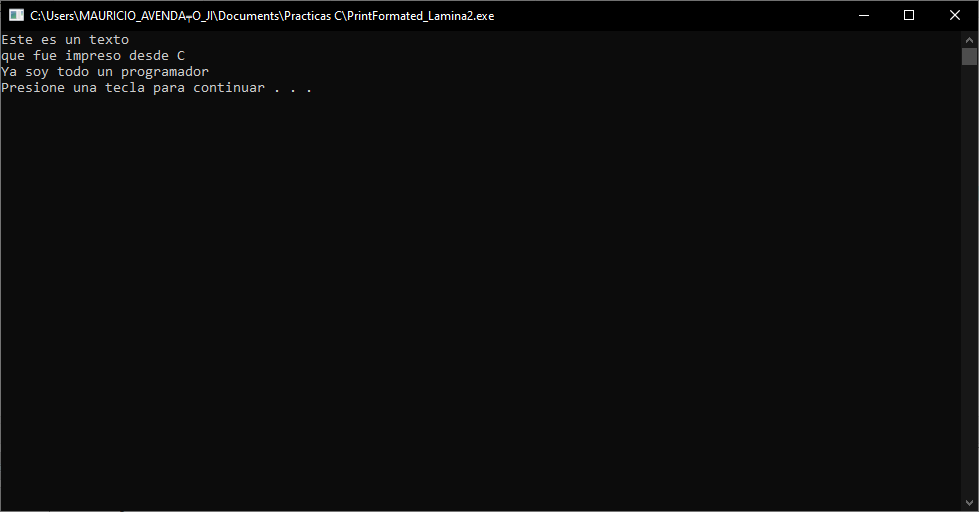
Texto que se va a imprimir en pantalla. La Secuencia de escape **\n** sirve para dar un salto de línea

Bibliotecas

Método principal

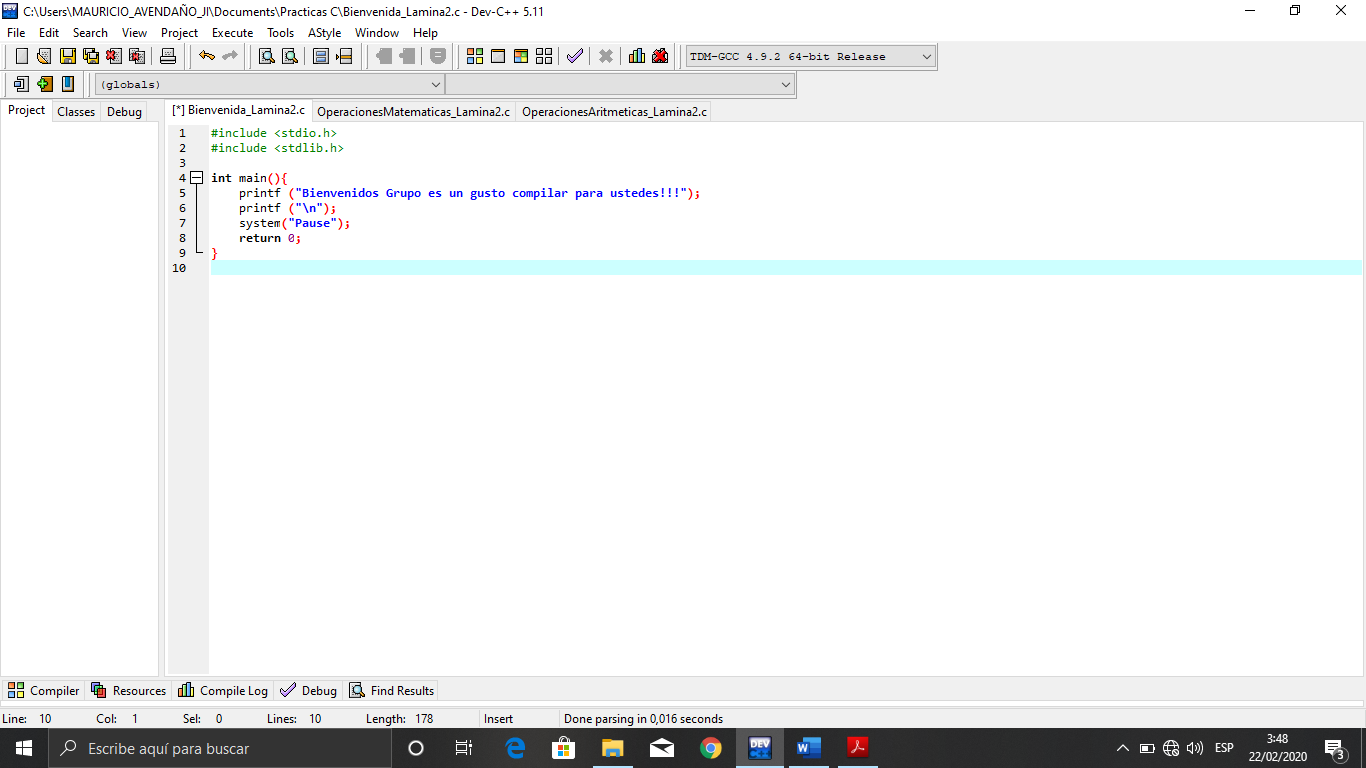
Valor que devuelve el método principal

Nota: El punto y coma (;) hace referencia que esa instrucción o línea de código se va a ir directamente a la memoria.



Ejercicio. Printf (Print formated)

Crea un archivo en Dev-C++, crea un programa que imprima una bienvenida.



Bibliotecas

Valor que devuelve el método principal

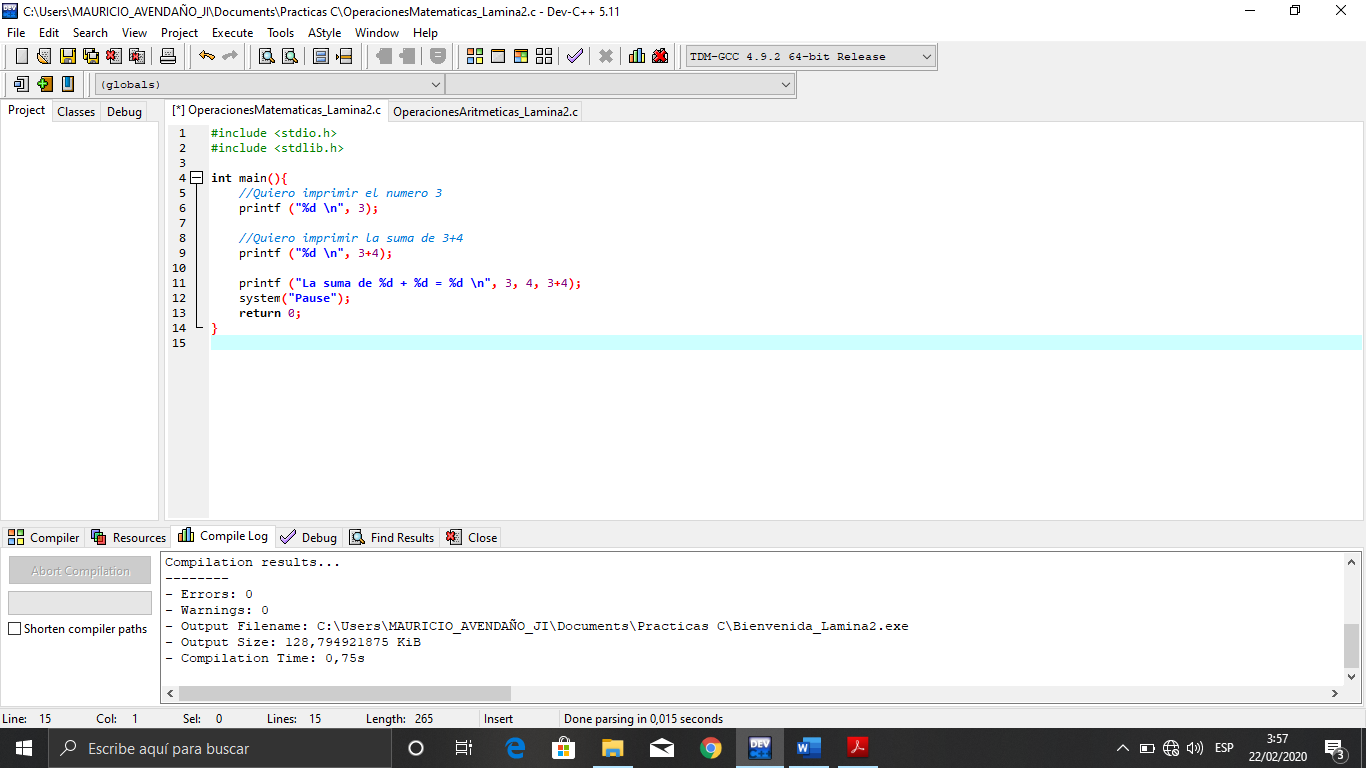
Función que nos permite detener la ejecución del programa y visualizar los resultados en pantalla

Método principal

Texto que se va a imprimir en pantalla. La Secuencia de escape **\n** sirve para dar un salto de línea



Ejemplo. Operaciones matematicas



Todo lo que este dentro del printf se va a imprimir en pantalla. La Secuencia de escape **\n** sirve para dar un salto de línea

Operaciones

Especificadores de conversión, en esta Secuencia se va a guardar el valor de la operación o variable

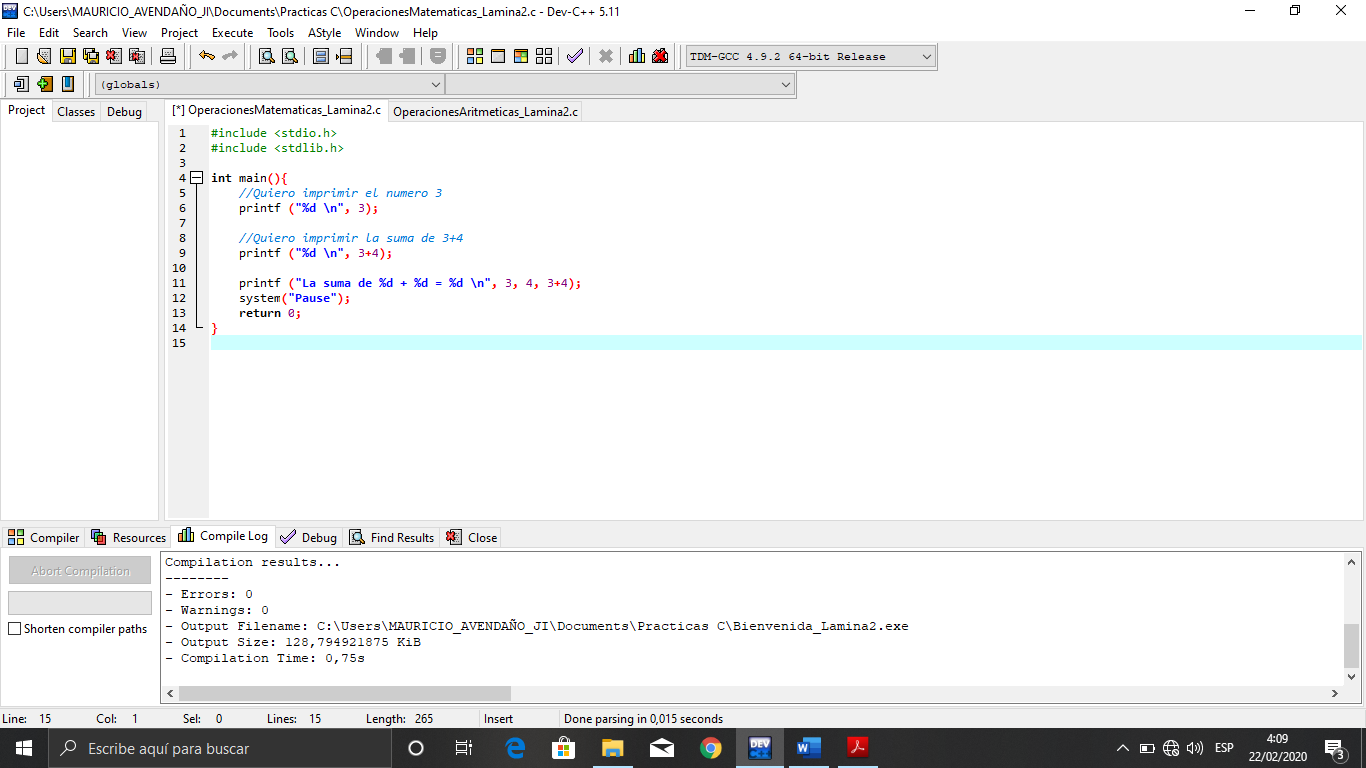
Función que nos permite detener la ejecución del programa y visualizar los resultados en pantalla

Método principal

Bibliotecas

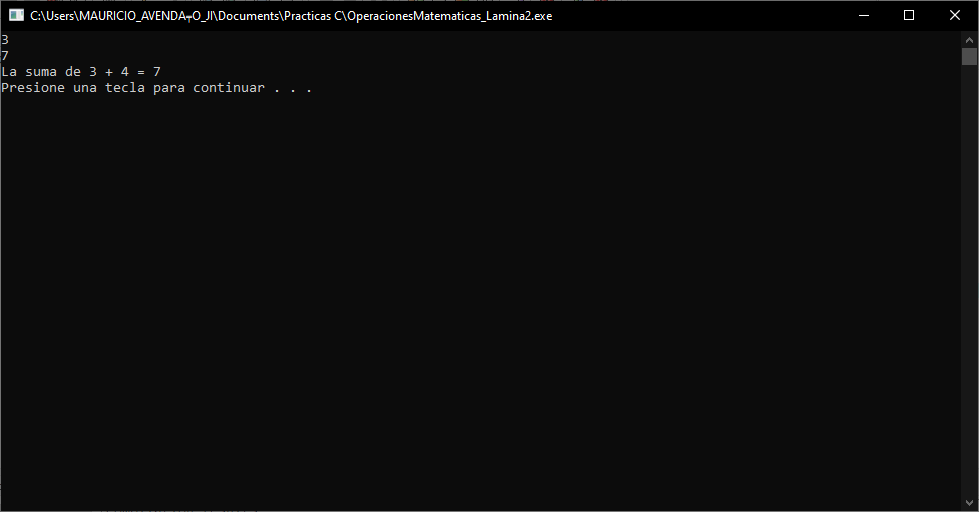
Valor que devuelve el método principal

Ejemplo:

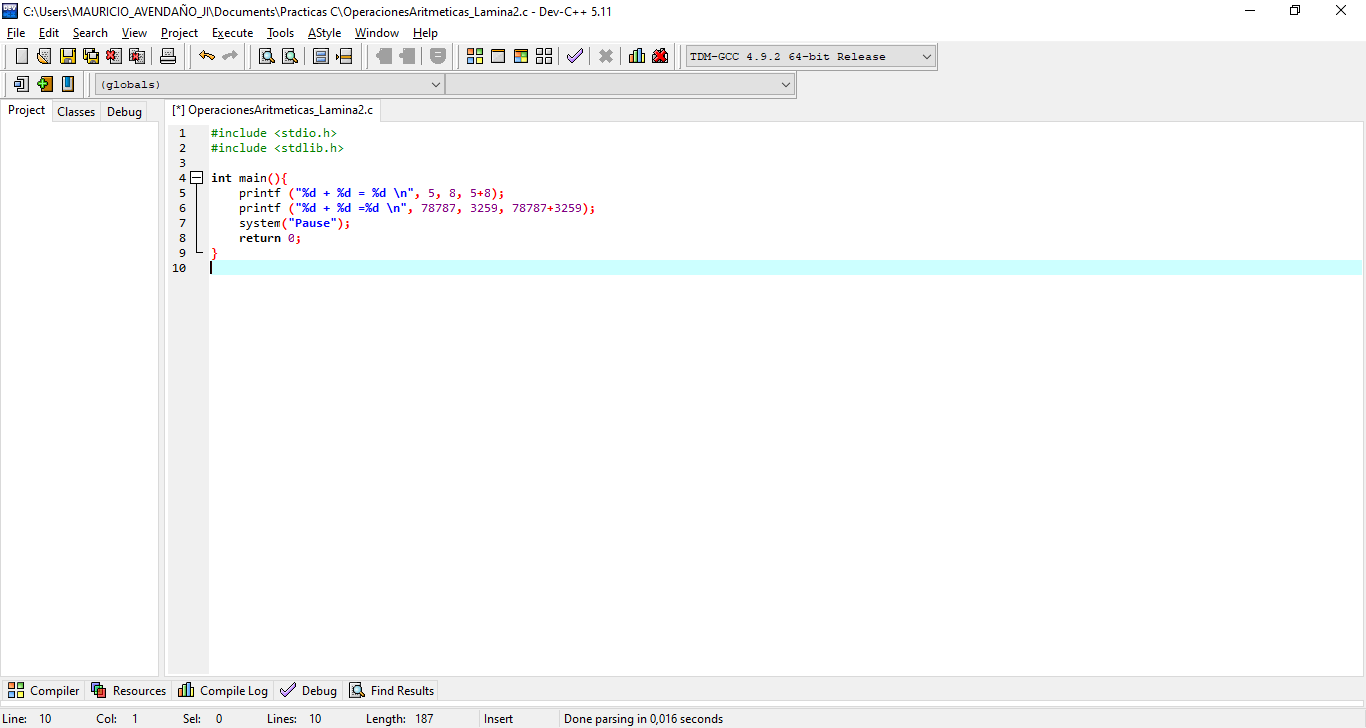


En esta parte se realiza la operación y se guarda en el especificador de acceso, pero puede ser un solo número, variable, etc., cabe mencionar que se pueden realizar más de una asignación, y por ende más de un especificador de conversión.

Esto es un especificador de conversión. Aquí se va a guardar el resultado de alguna operación, un valor, el valor de una variable, etc. dependiendo del tipo de dato que sea ese valor.



Ejercicio. Operaciones matemáticas. Crea un archivo en Dev-C++ que se llame Operaciones Aritméticas. En el método principal (main) escribir el código para imprimir la suma de 5+8, 78787+3259.



Especificadores de conversión.

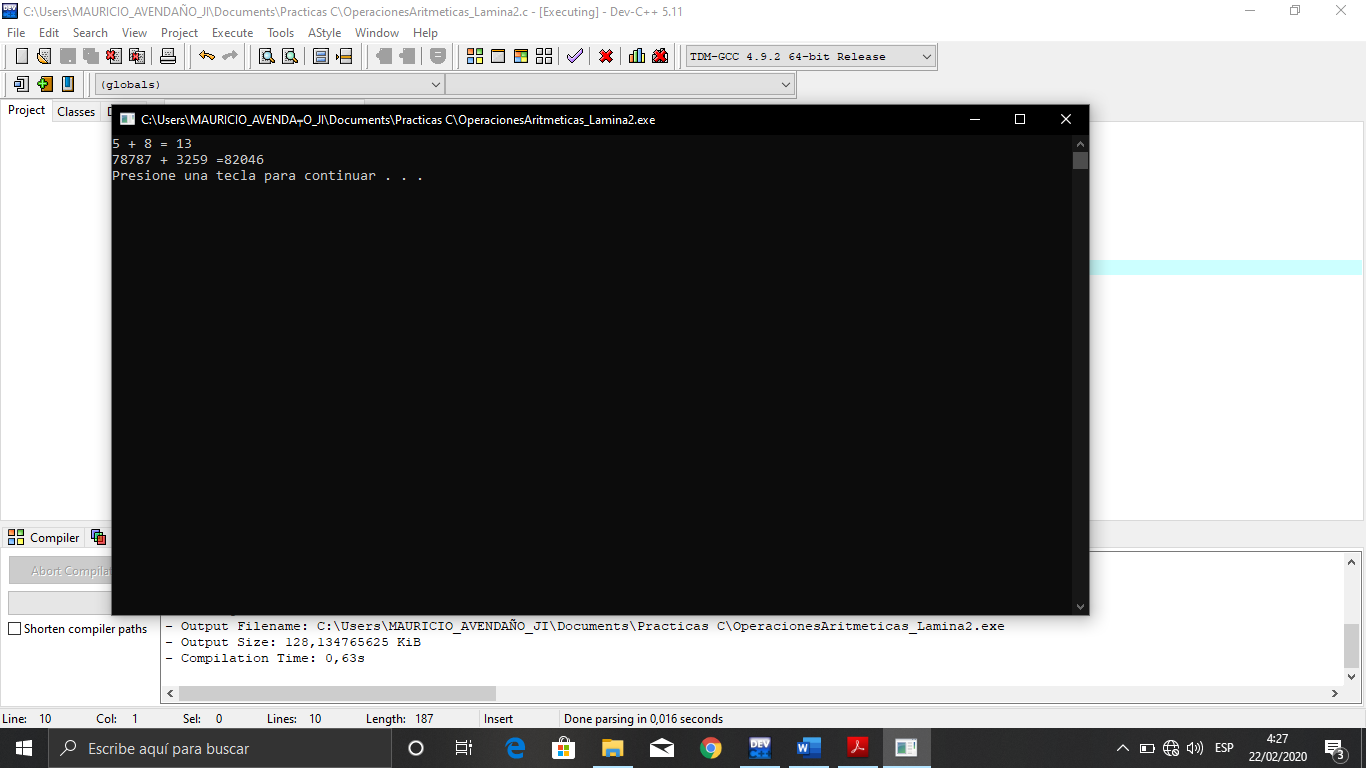
Operaciones

Método principal

Bibliotecas

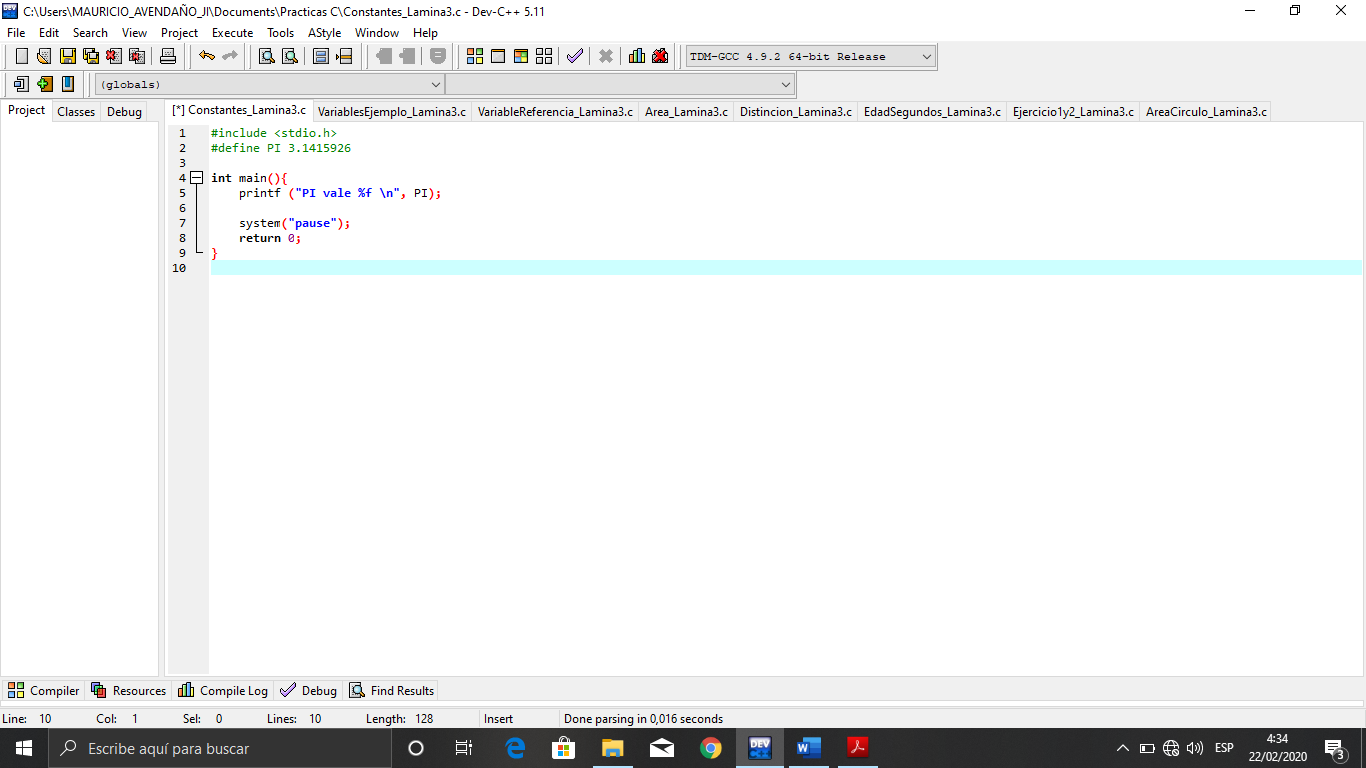
Función que nos permite detener la ejecución del programa y visualizar los resultados en pantalla

Valor que devuelve el método principal



**Documento 3**

Ejemplo. Uso de constantes



Asignación de la constante al especificador

Especificador de conversión

Función que nos permite detener la ejecución del programa y visualizar los resultados en pantalla

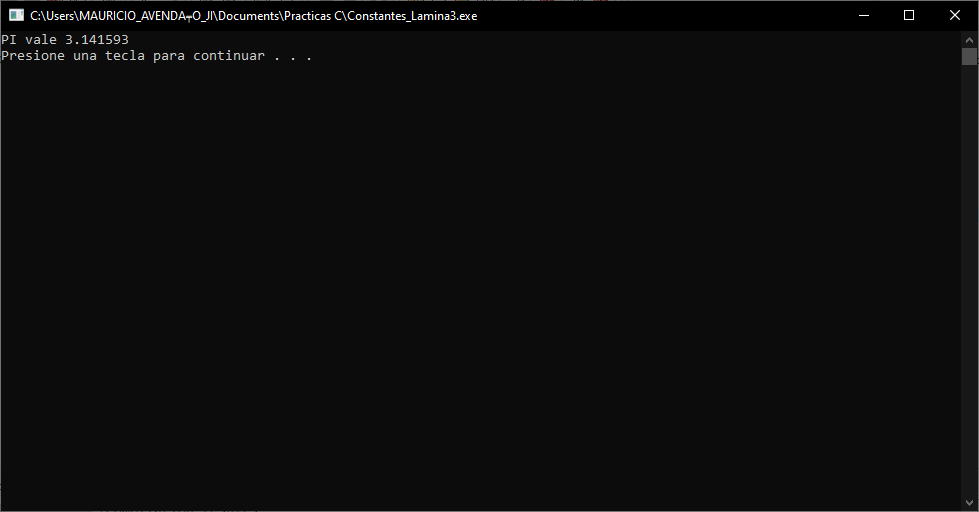
Mensaje

Se declara constante

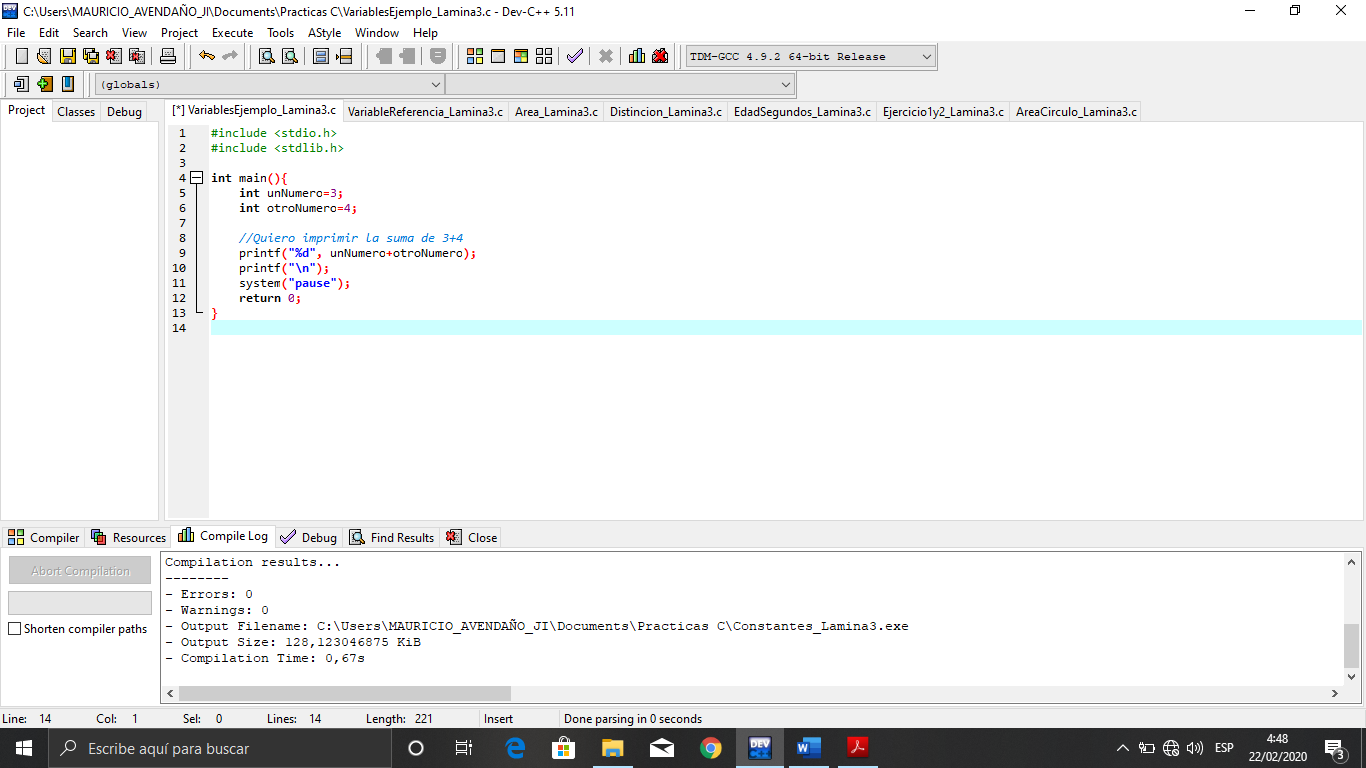
Bibliotecas

Método principal

Valor que devuelve el método principal



Ejemplo. Uso de variables



Secuencia de escape que sirve para dar un salto de línea

Especificador de conversión

Suma de las variables unNumero más otroNumero (3+4=7).

Comentario

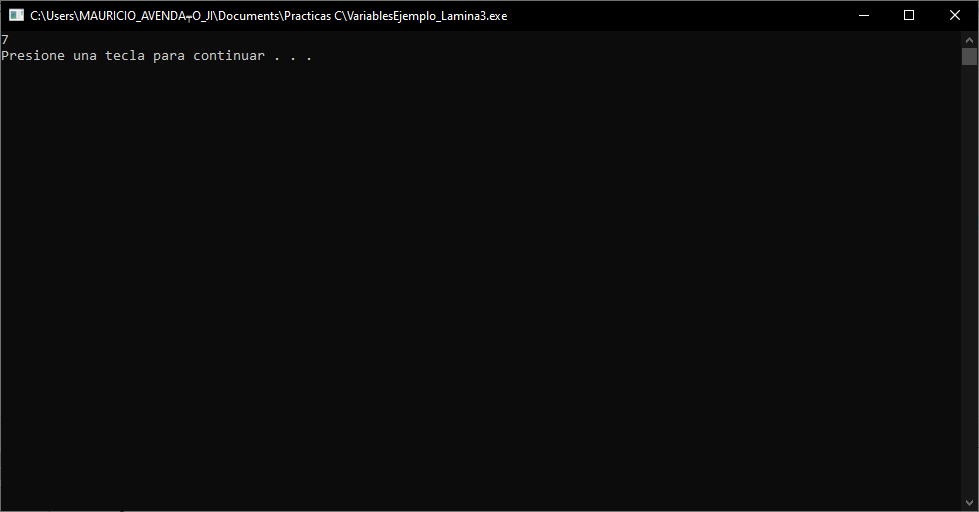
Declaración de variables con su respectivo valor

Función que nos permite detener la ejecución del programa y visualizar los resultados en pantalla

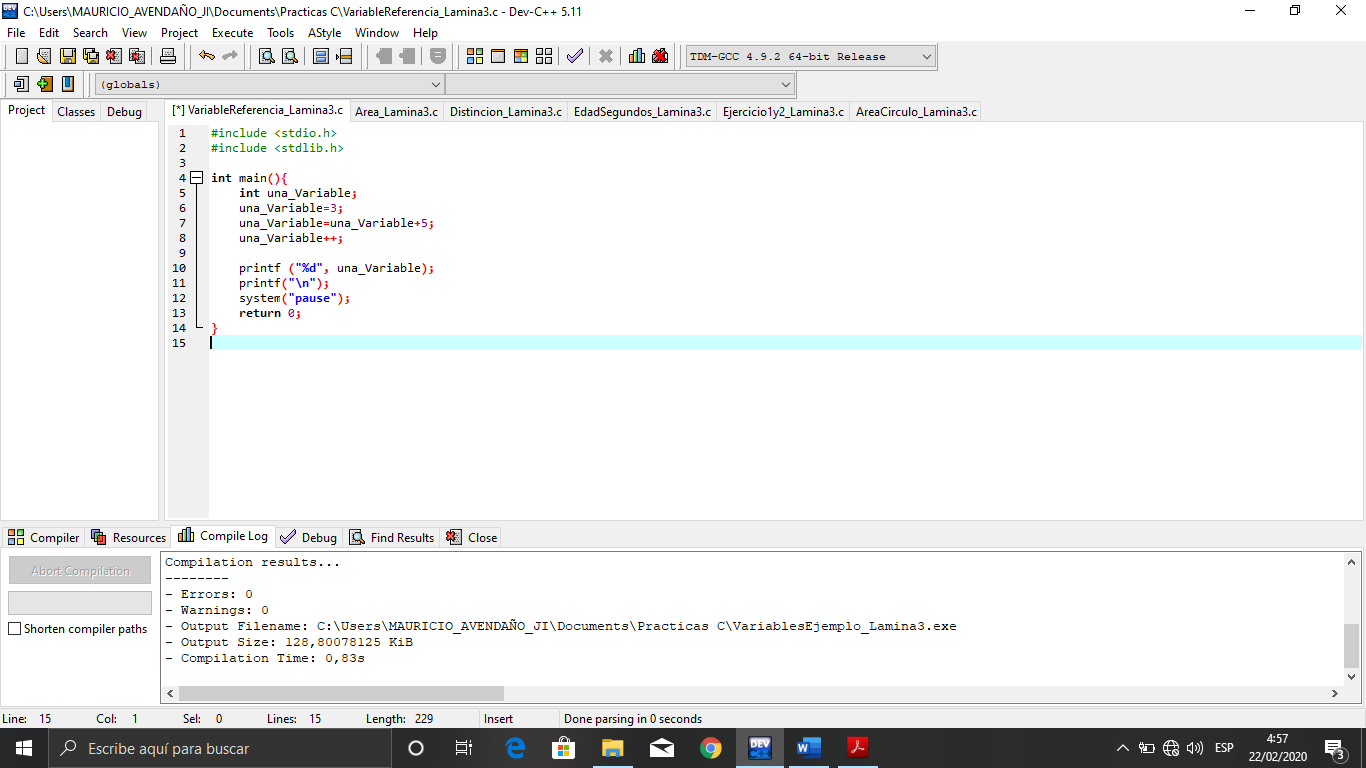
Método principal

Bibliotecas

Valor que devuelve el método principal



Ejemplo. Uso de variables



Al inicio la variable tiene el valor de 3, después le suma 5, valiendo 8, después la incrementa y ahora vale 9.

impresión de la variable, primero nos encontramos con el especificador de conversión y luego la asignación de la variable

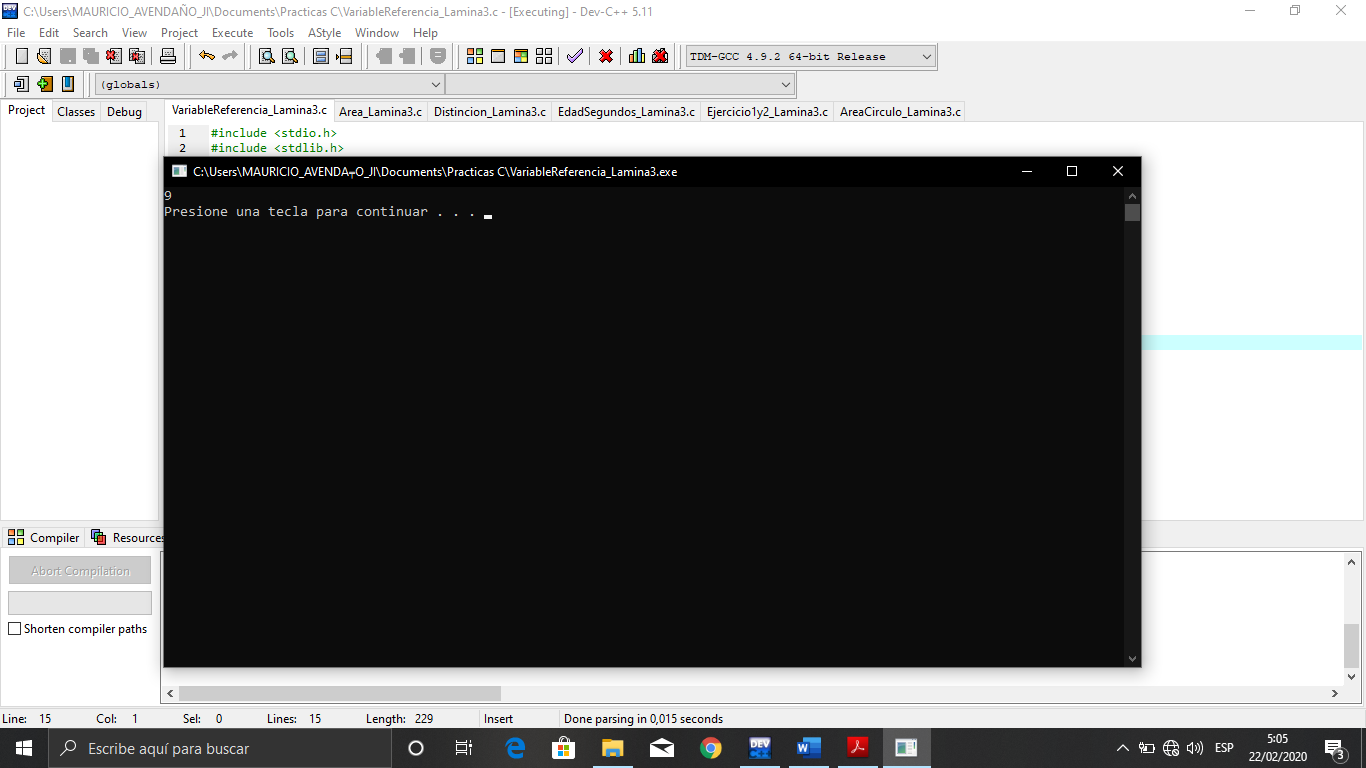
Método principal

Declaración de la variable una\_Variable

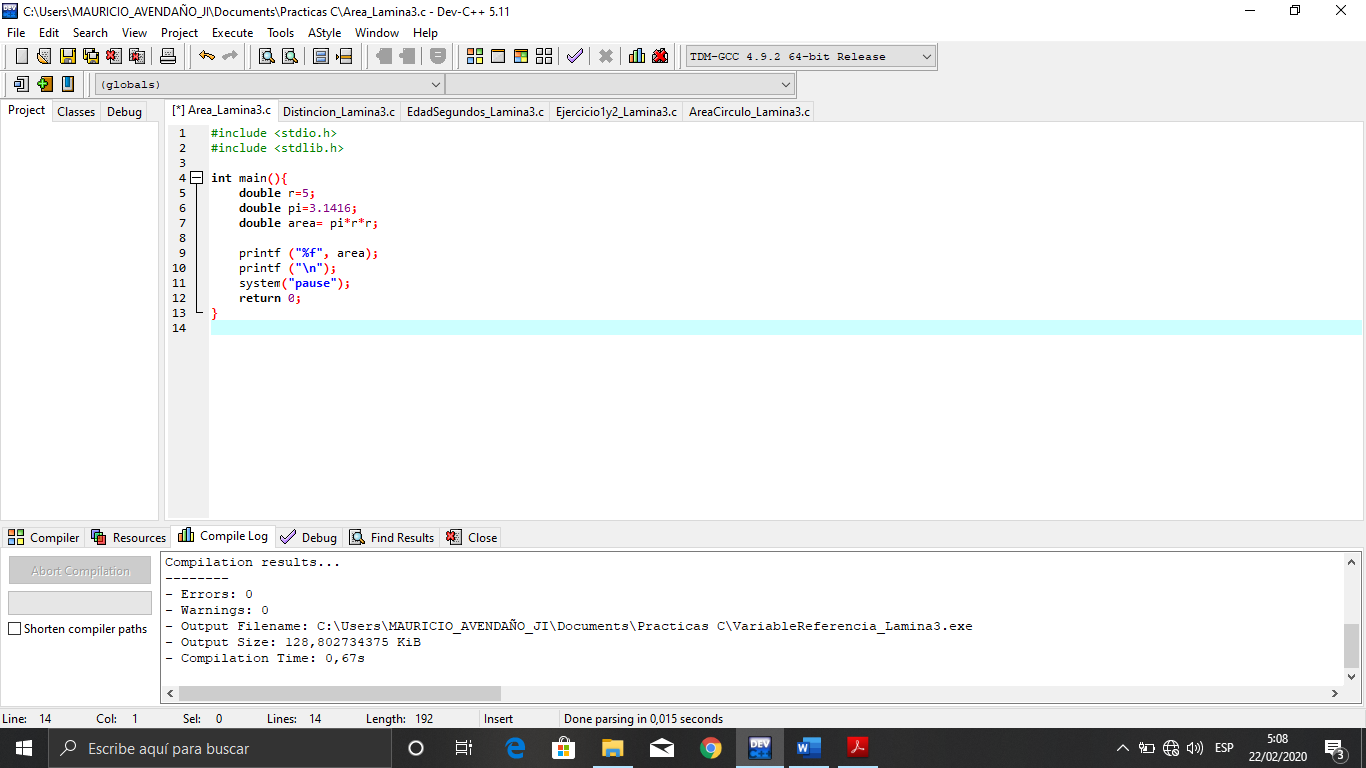
Función que nos permite detener la ejecución del programa y visualizar los resultados en pantalla

Bibliotecas

Valor que devuelve el método principal



Ejemplo. Variables



Impresión de un salto de linea

Impresión de la variable área, primero se encuentra el especificador de conversión en el cual se va a guardar el valor de área

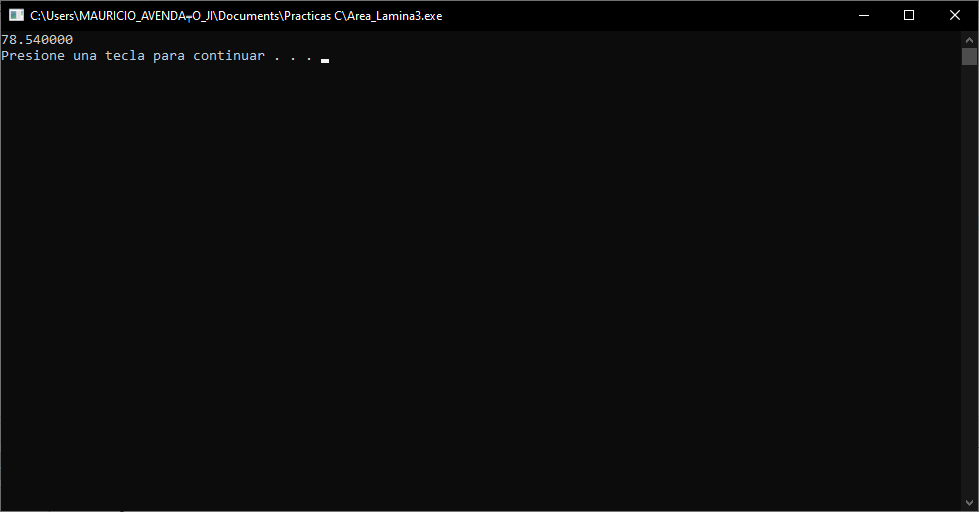
Declaración de variables con tipo de dato double con un valor asignado, área realiza la operación para calcular el área

Función que nos permite detener la ejecución del programa y visualizar los resultados en pantalla

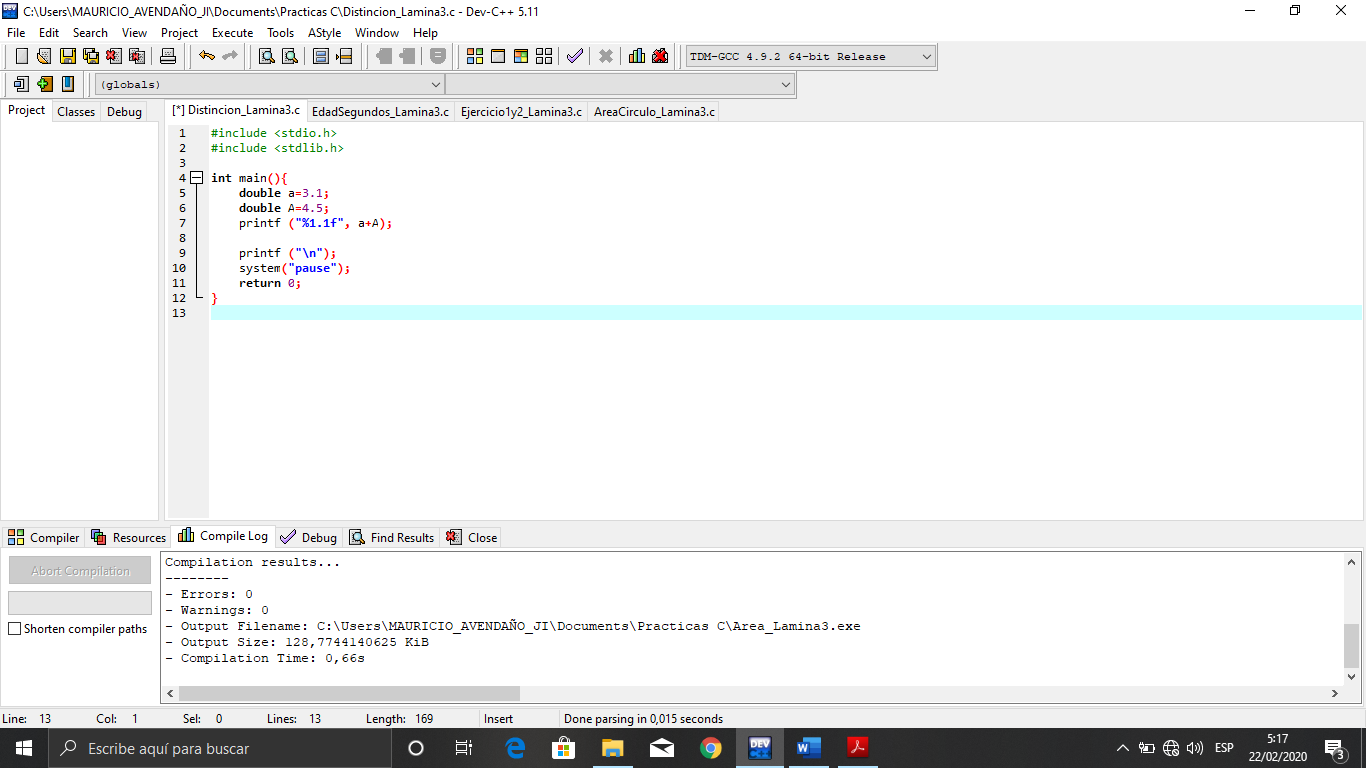
Método principal

Bibliotecas

Valor que devuelve el método principal



Ejemplo. Variables. C distingue entre mayúsculas y minúsculas



Especificador de conversión, espacio donde se guarda un valor.

Aquí se suma a+A, 3.1+4.5, el resultado se guarda en el especificador de conversión y le indica que imprima solo un decimal

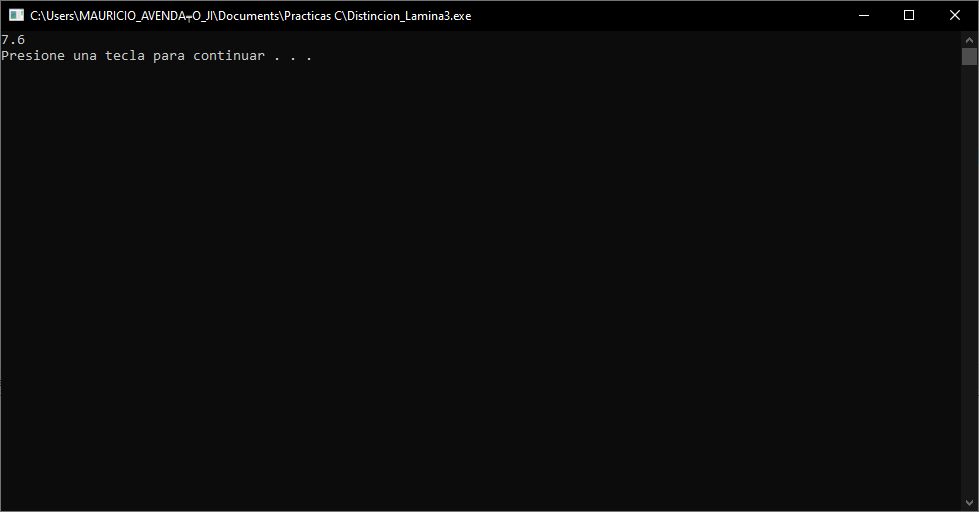
Método principal

Declaración de variables con tipo de dato double con un valor asignado.

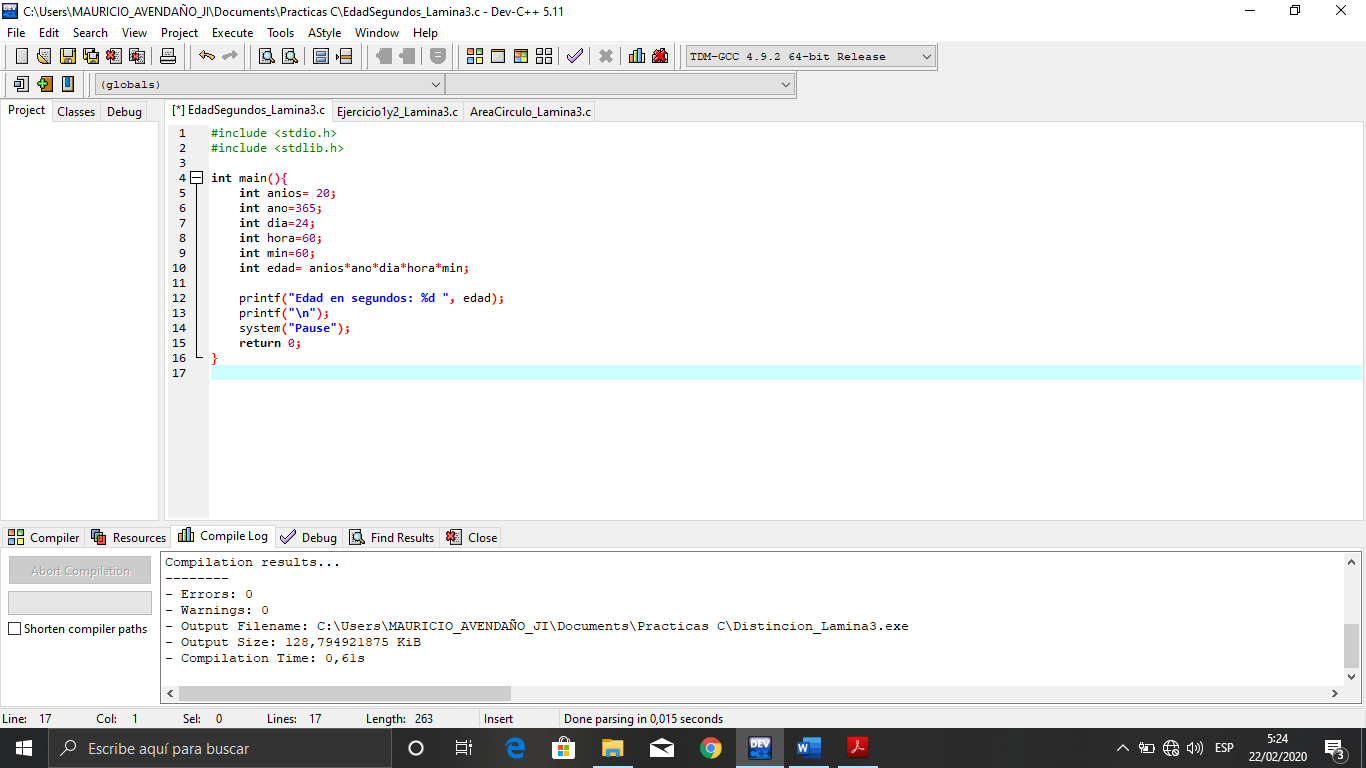
Función que nos permite detener la ejecución del programa y visualizar los resultados en pantalla

Valor que devuelve el método principal

Bibliotecas



Ejercicio. Variables. Calcula la cantidad de segundos que has vivido



Especificador de conversión, espacio donde se guarda el valor de edad, con un mensaje, el tipo de dato que esta recibiendo es entero

Aquí se multiplican los años de edad, por el valor de ano, día, hora, minutos y el resultado se guarda en edad, luego edad se guarda en el especificador de conversión.

Declaración de variables con tipo de dato double con un valor asignado.

Bibliotecas

Método principal

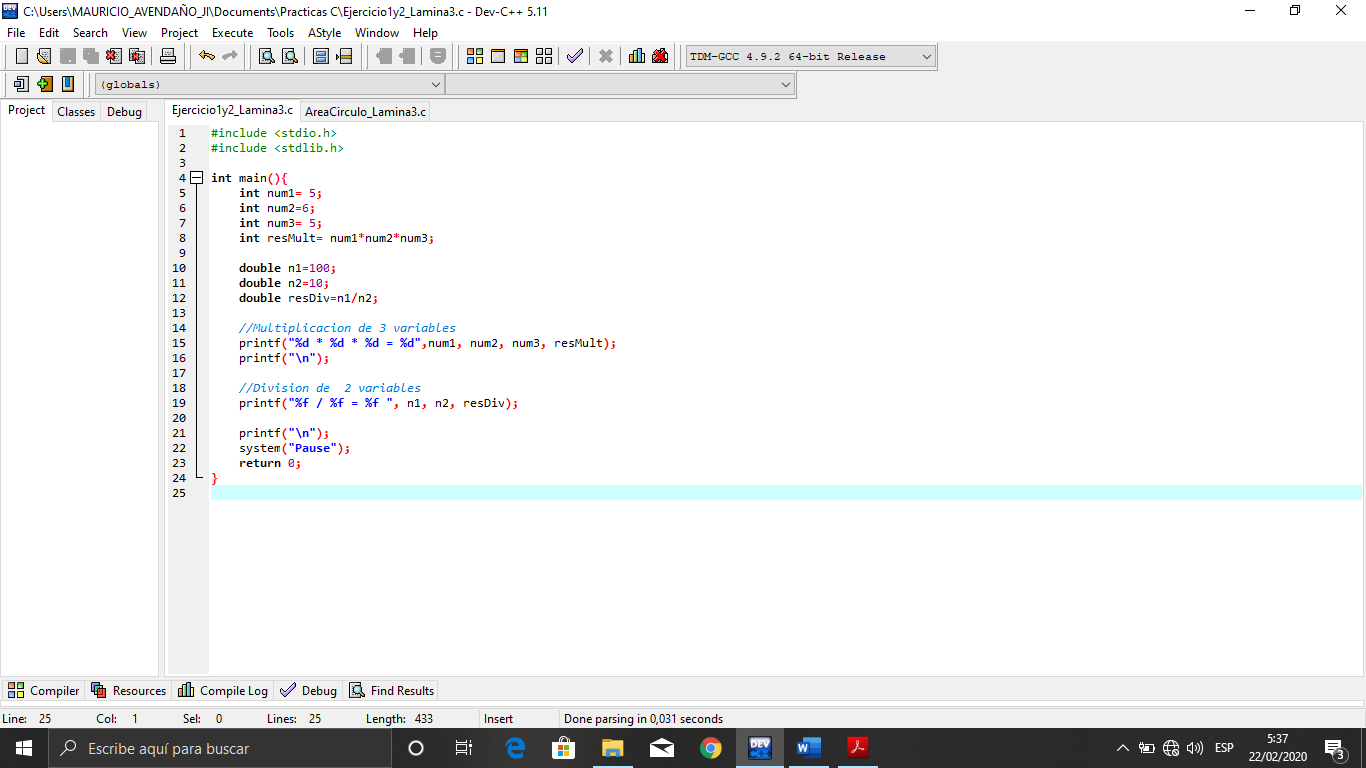
Función que nos permite detener la ejecución del programa y visualizar los resultados en pantalla

Valor que devuelve el método principal



Ejercicio. Operadores aritméticos. 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 cero, mostrar el resultado de dividir la primera entre la segunda.



Recibimos cuatro valores y los asignamos a los especificadores de acceso dependiendo del orden, en el primero el num1, el segundo el num2, el tercero el num3, y en cuarto que es el resultado, resMult.

Esto mismo sucede para la impresión de la división, solo que aquí solo son tres valores esperados, el de n1, n2, y resDiv

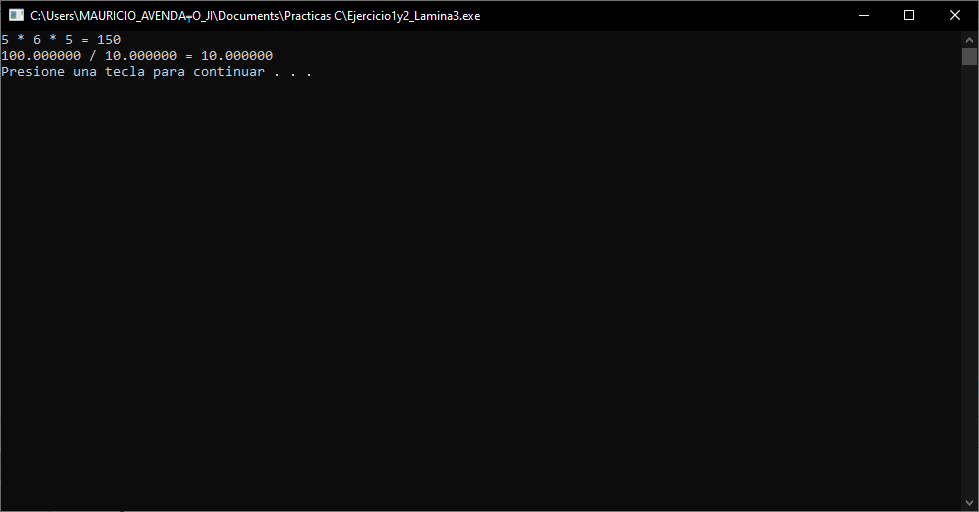
Declaración de variables y su valor, las primeras son para la multiplicación y la segunda para la división, el primer resultado se guarda en resMult y el segundo en resDiv

Función que nos permite detener la ejecución del programa y visualizar los resultados en pantalla

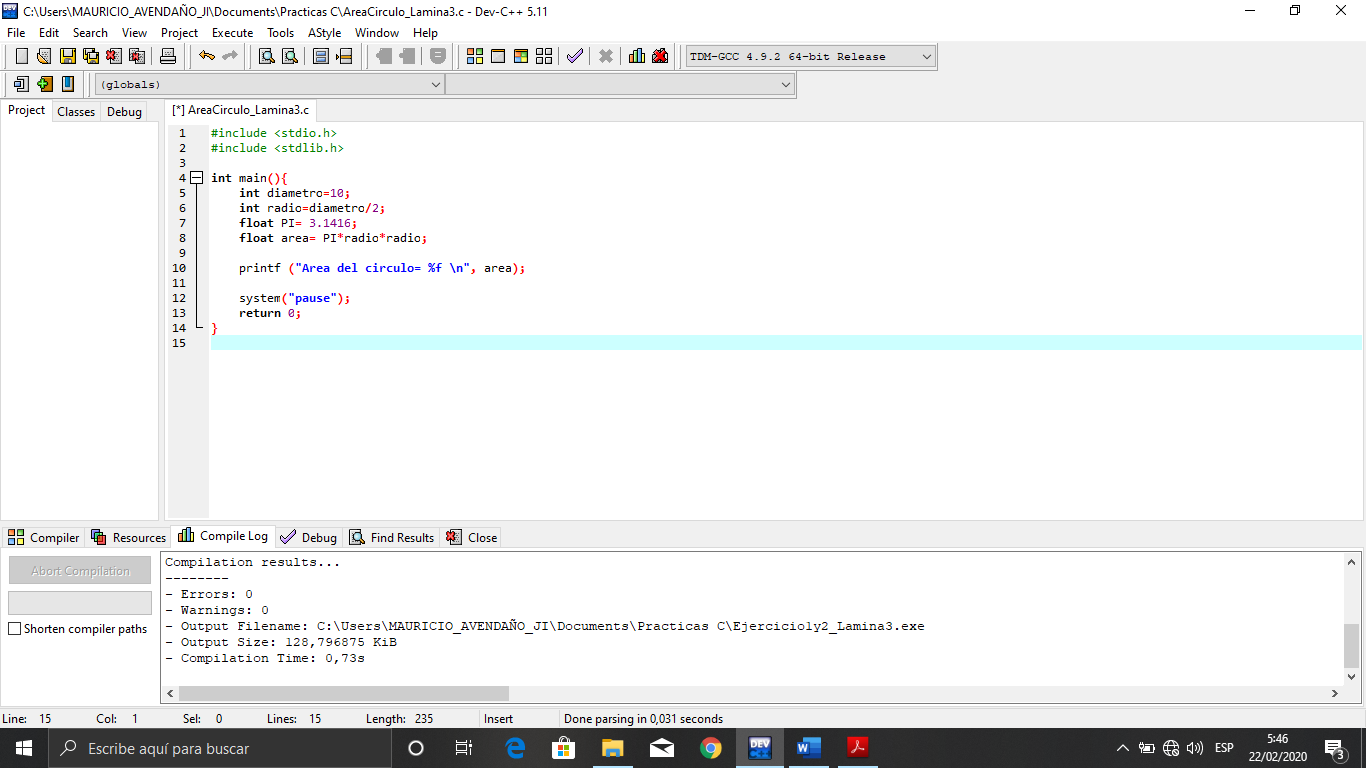
Valor que devuelve el método principal

Método principal

Bibliotecas



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



aquí se hace la impresión del mensaje: área del circulo y en el especificador de conversión se guarda el valor de área, el tipo de dato a recibir es de tipo flotante, área es la asignación.

Declaración de las variables diámetro=10, radio, pi y área, necesarias para calcular el área, la formula es πr2.

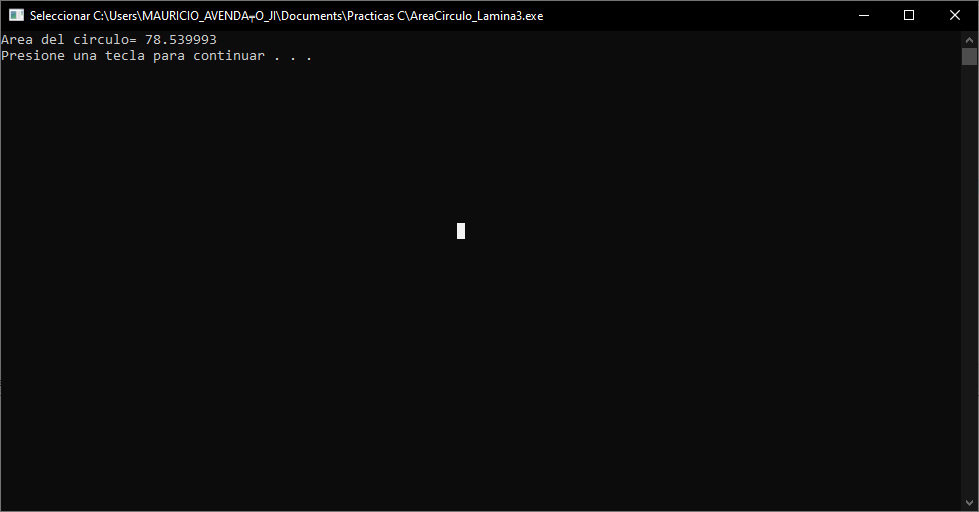
Cada uno con su respectivo valor, en área se guarda el resultado después de aplicar la formula.

Método principal

Bibliotecas

Función que nos permite detener la ejecución del programa y visualizar los resultados en pantalla

Valor que devuelve el método principal



**Documento 4**

Ejemplo. Operador POST incremento/decremento. El operador POST toman el valor de la variable y luego la incrementan o decrementan.



Comentario, este no afecta la ejecución del programa.

Especificador de conversión

En este punto, tomamos el valor de la variable var (1) y lo asignamos al especificador de conversión, luego se incrementa (1+1=2), después imprimimos a la misma variable, pero con su nuevo valor.

Lo mismo sucede con la variable con, pero en vez de incrementar decrementa, es decir, le resta 1, primero vale 1, se asigna al especificador de conversión, luego le resta 1, y ahora vale 0, este se imprime en la siguiente instrucción, cuando se asigna el otro especificador de conversión.

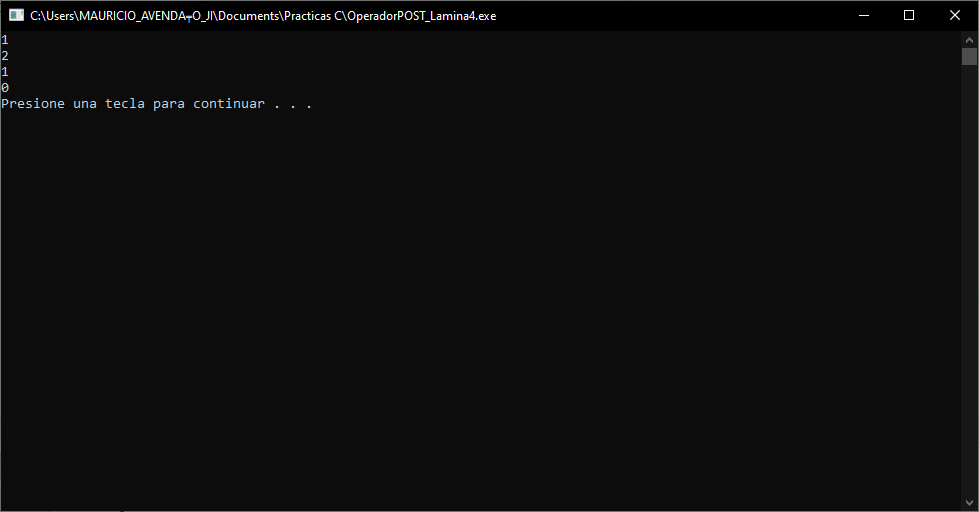
Declaración de las variables var y con, cada una de ellas inicializada en 1, ambas de tipo entero

Método principal

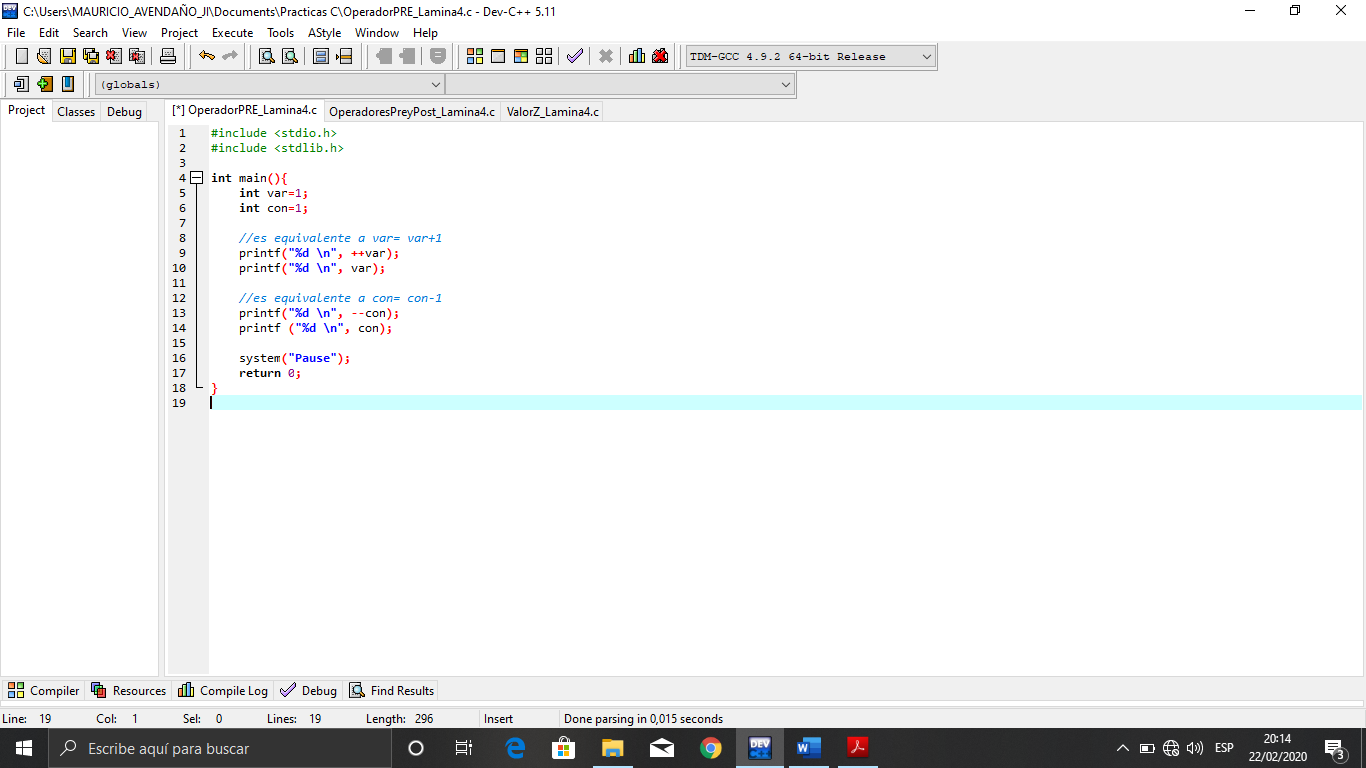
Función que nos permite detener la ejecución del programa y visualizar los resultados en pantalla

Valor que devuelve el método principal

Bibliotecas



Ejemplo. Operador PRE incremento/decremento. Operador PRE, primero se incrementa o decrementa el valor de la variable y después hace uso de ella.



Variables

Aquí, primero incrementamos a var (1+1=2), la asignamos al especificador, después tomamos el valor de la misma e igual se imprime, lo mismo pasa con, primero la decrementa (1-1=0), asigna el resultado al especificador y después la imprime.

Especificador de conversión

Comentario, este no afecta la ejecución del programa.

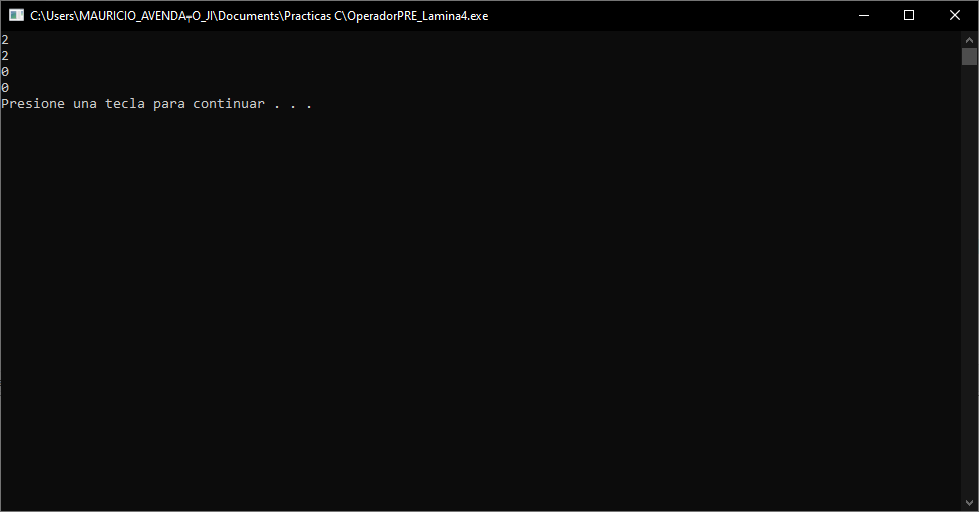
Función que nos permite detener la ejecución del programa y visualizar los resultados en pantalla

Declaración de las variables var y con, cada una de ellas inicializada en 1, ambas de tipo entero

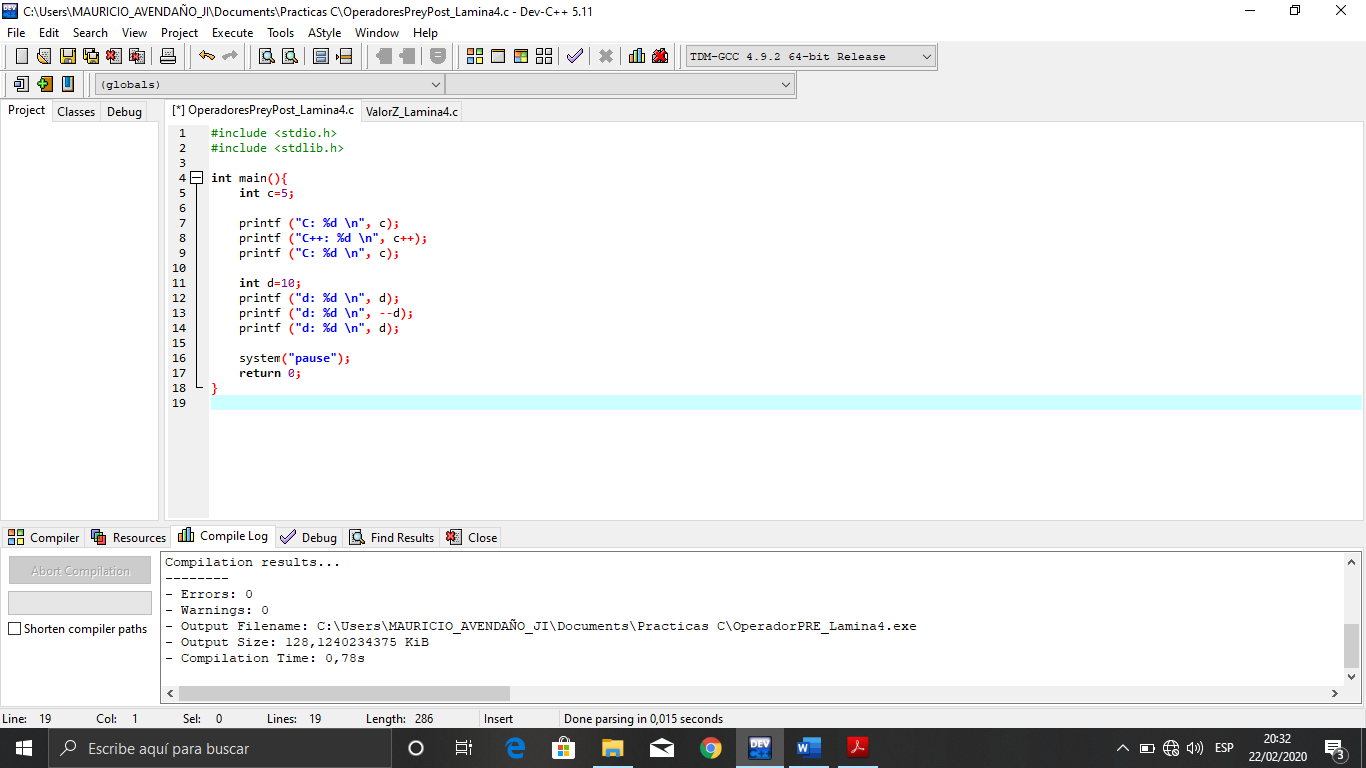
Método principal

Bibliotecas

Valor que devuelve el método principal



Ejemplo. Operadores incremento y decremento.



Variable llamada c de tipo entero con valor 5

Proceso:

Primero c vale 5, lo asigna en el especificador de conversión y lo imprime, luego se asigna de nuevo y se imprime el mismo valor, pero después incrementa, ahora c vale 6, y ese valor se asigna y se imprime.

En la otra parte sucede algo similar, primero d vale 10, se asigna al especificador de conversión y se imprime, luego se decrementa, ahora vale 9, se asigna y se imprime, y al último se asigna el mismo valor y se imprime.

%d🡺especificador de conversión

\n🡺salto de línea

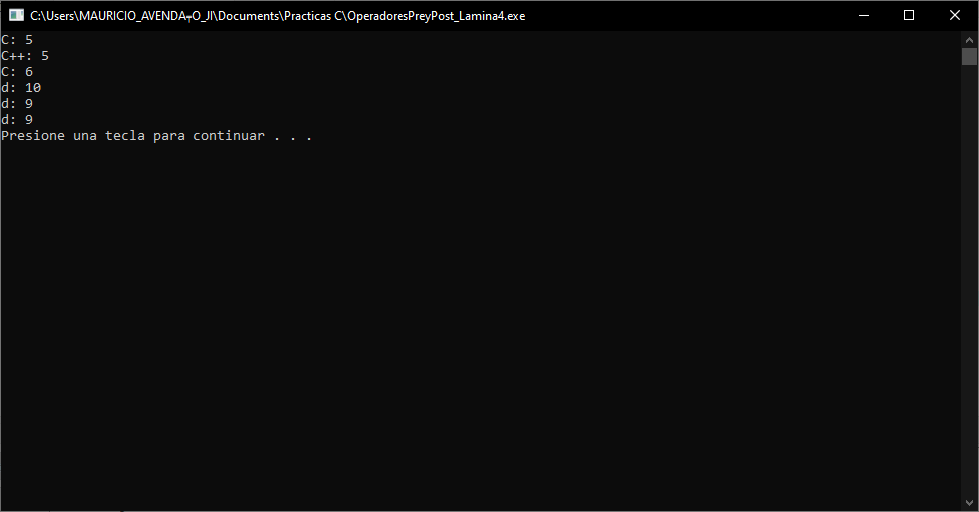
Variable llamada d de tipo entero con valor 10

Función que nos permite detener la ejecución del programa y visualizar los resultados en pantalla

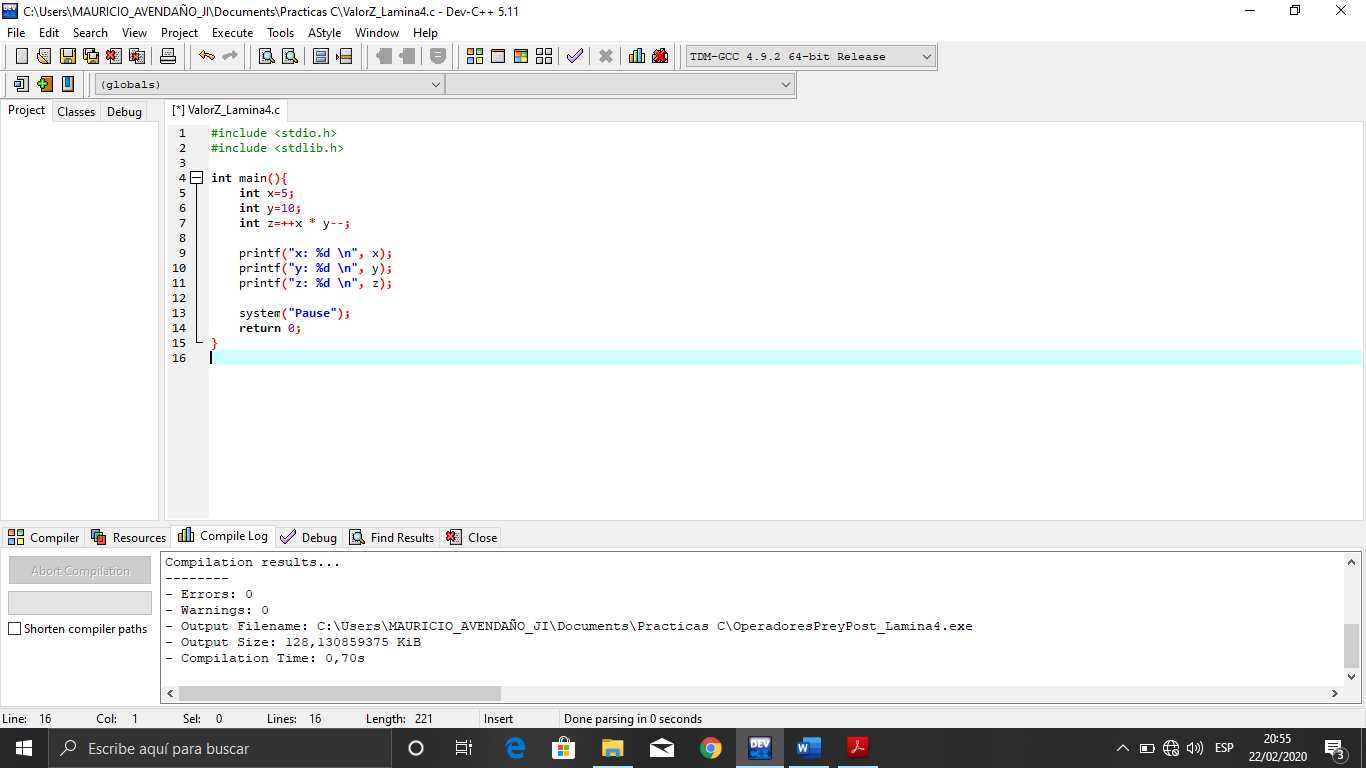
Método principal

Bibliotecas

Valor que devuelve el método principal



Ejercicio. operadores incremento/decremento. Determina cual es el valor de z



X= 6, primero valía 4 pero se le sumo 1.

Y=9, primero valía 10, pero se le resto 1

Z= 6\*10, porque x si incremento, pero y, primero tomo su valor y luego lo decremento

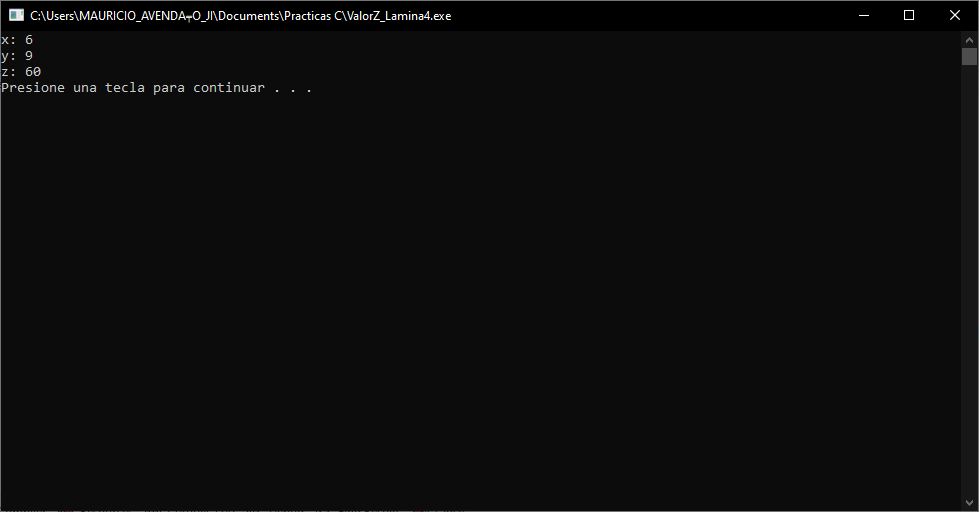
En C necesitamos declarar las variables a usar, por lo que aquí estamos haciendo ese proceso, declaramos las variables x=5, y=10 y z es igual a x+1 por y, ya que, y primero lo multiplica y luego lo decrementa, pero cuando se vaya a usar de nuevo a y, entonces su valor ya cambia.

Método principal

Función que nos permite detener la ejecución del programa y visualizar los resultados en pantalla

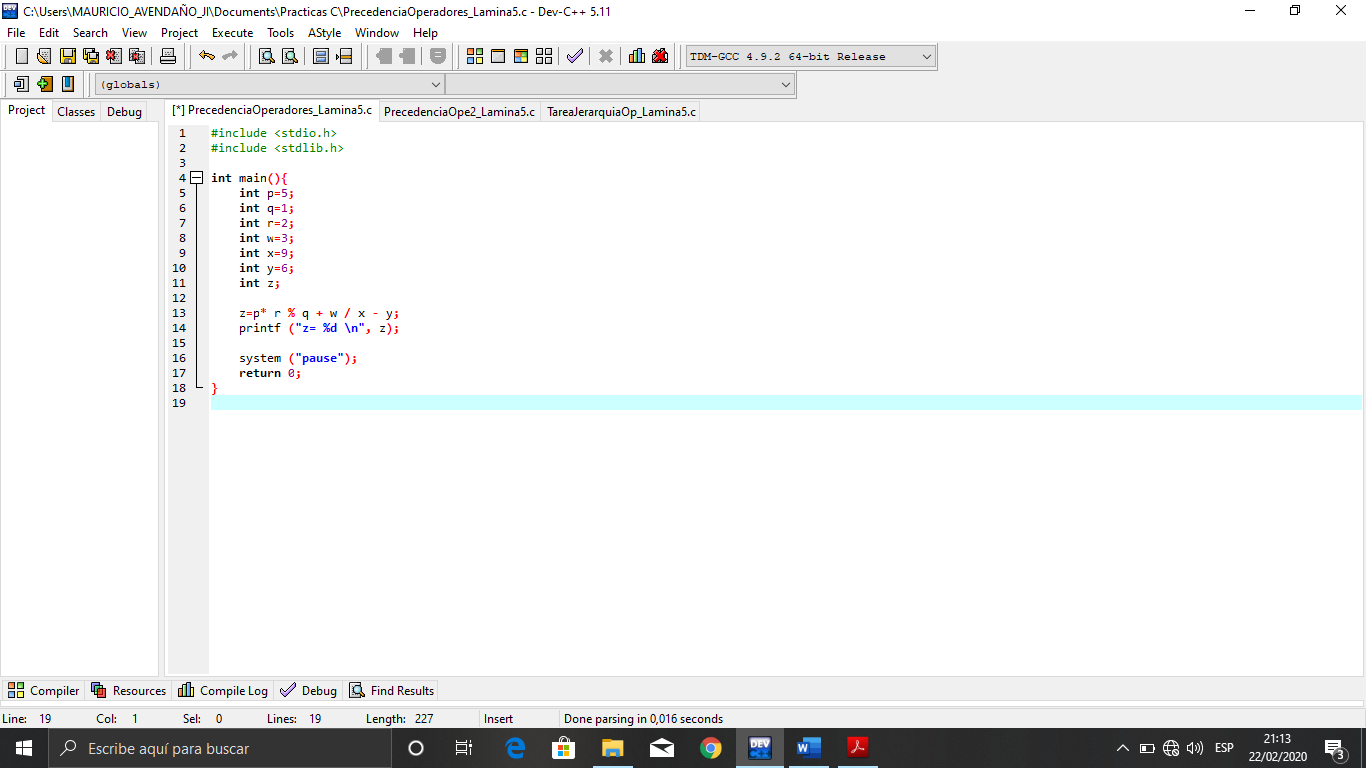
Valor que devuelve el método principal

Bibliotecas



**Documento 5**

Ejemplo. Precedencia de operadores.



El valor de z se asigna al especificador de conversión (%d) y se va a imprimir en pantalla y al final sara un salto de linea.

Aquí se hacen las operaciones correspondientes para determinar el valor de z y el resultado de estas se guarda en la variable z.

Declaración de las variables con su respectivo valor, todas de tipo entero. Para declarar una variable: tipoDato nombre;

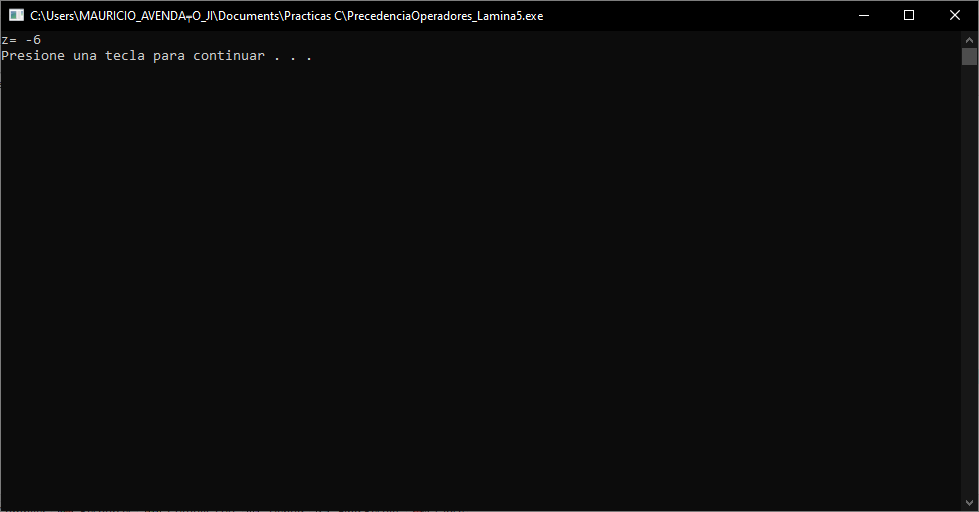
El punto y coma hace referencia que esa línea se va a guardar en memoria.

Función que nos permite detener la ejecución del programa y visualizar los resultados en pantalla

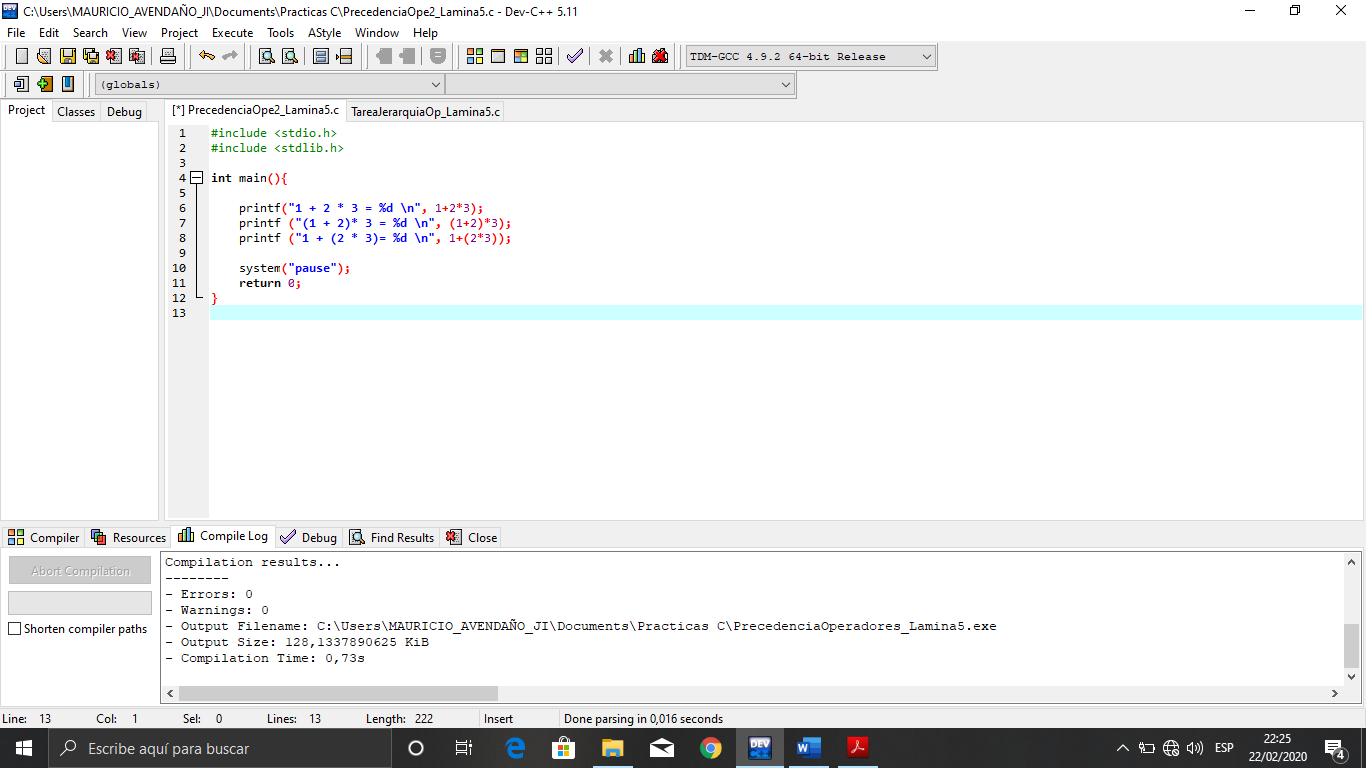
Método principal

Bibliotecas

Valor que devuelve el método principal



Ejemplo. Precedencia de operadores.



Todo aquello que se encuentre entre comillas será impreso en pantalla, excepto las Secuencias de escape (\n), y los especificadores de conversión, la función de estos es guardar un valor e imprimir ese valor mas no la Secuencia tal cual.

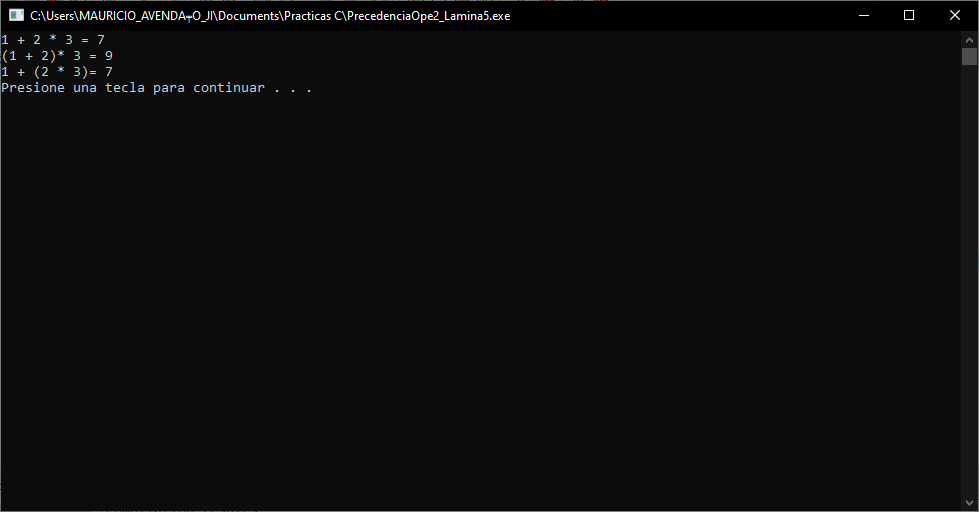
Operaciones, el resultado arrojado se va a asignar al especificador de conversión (%d).

Método principal

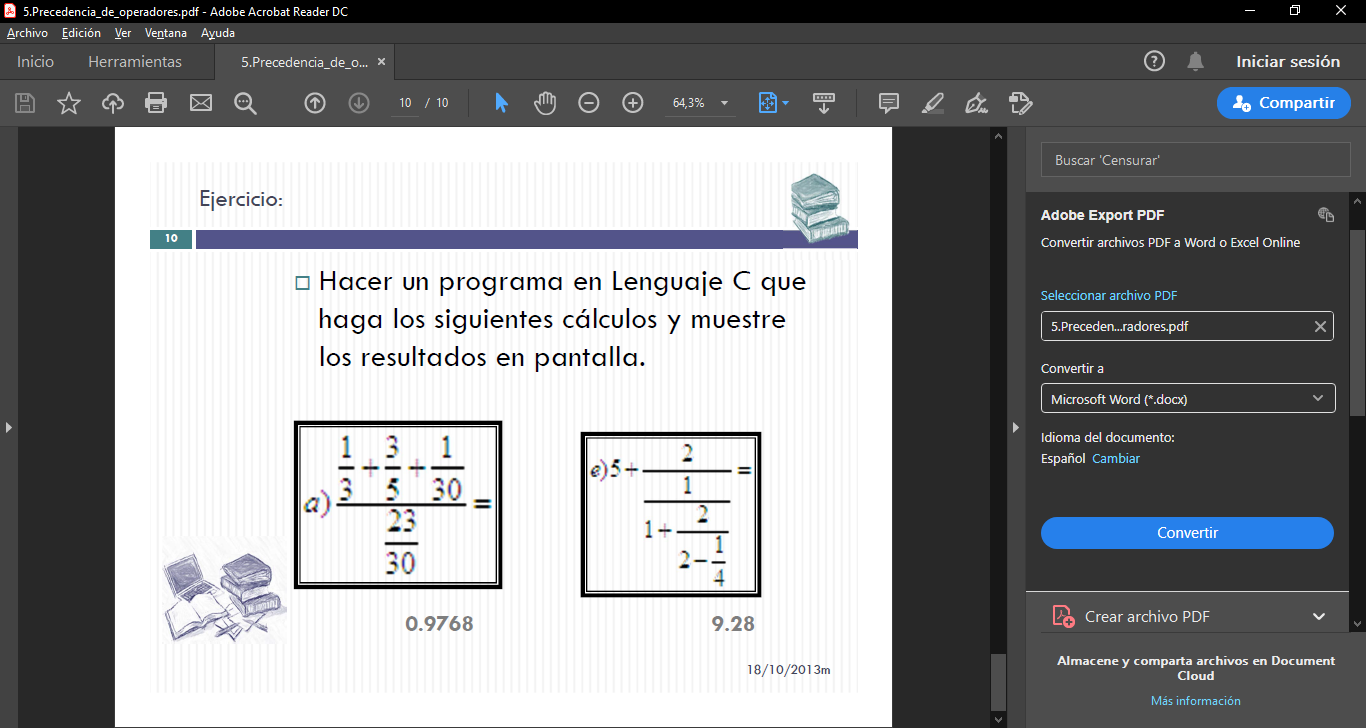
Función que nos permite detener la ejecución del programa y visualizar los resultados en pantalla

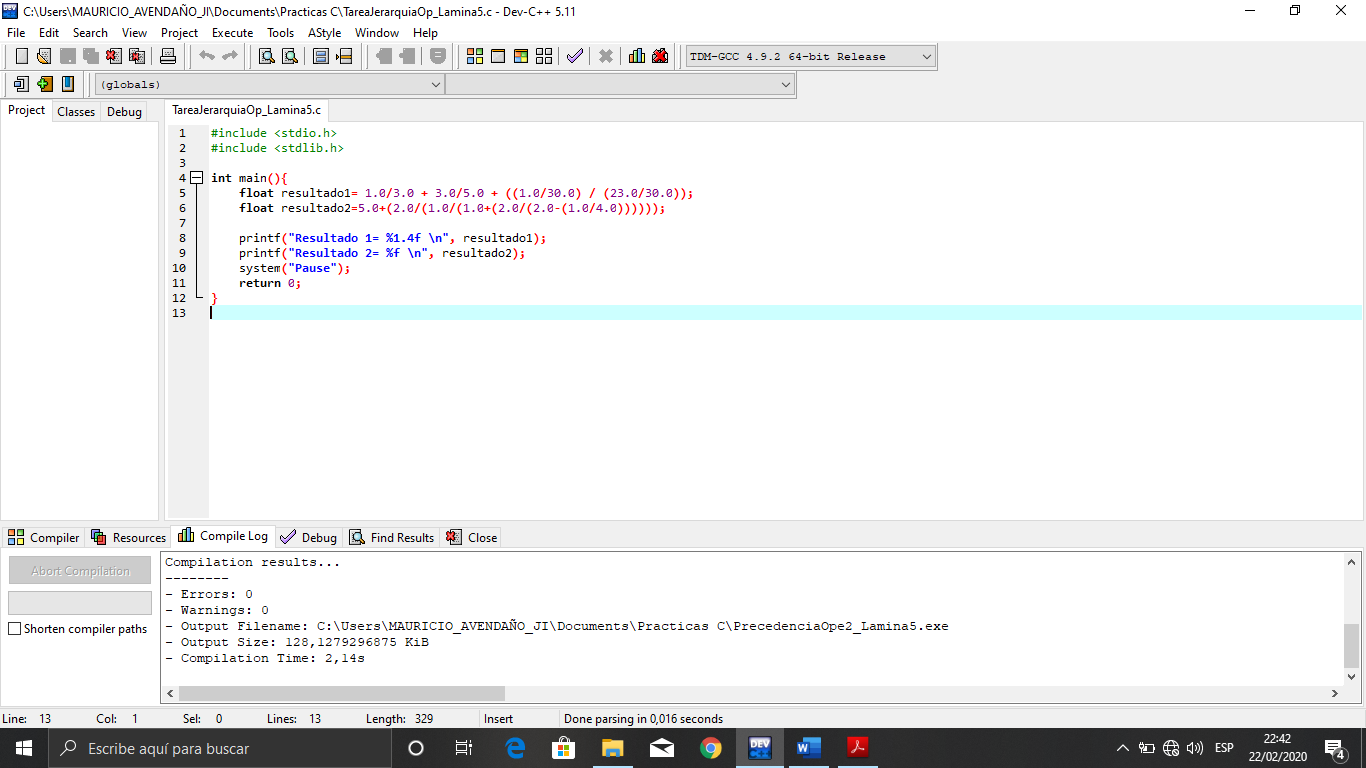
Bibliotecas

Valor que devuelve el método principal



Ejercicio. Precedencia de operadores. Hacer un programa en lenguaje C que haga los siguientes cálculos y muestre los resultados en pantalla.





Bibliotecas

Método principal

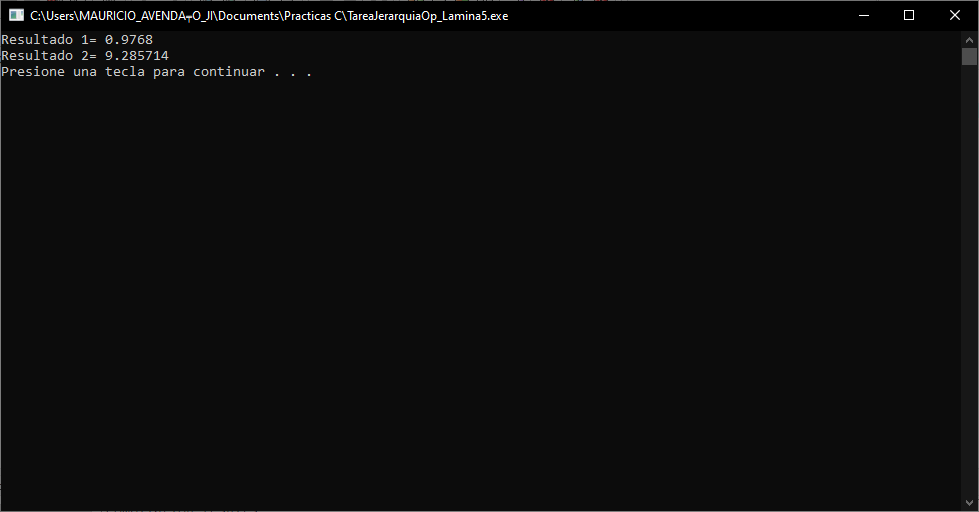
Aquí se declaran las variables de tipo de dato entero, así mismo, se realizan las operaciones para que ahí mismo se guarde el resultado de cada una.

Es importante tomar en cuenta la jerarquía de los operadores para que el resultado sea el esperado.

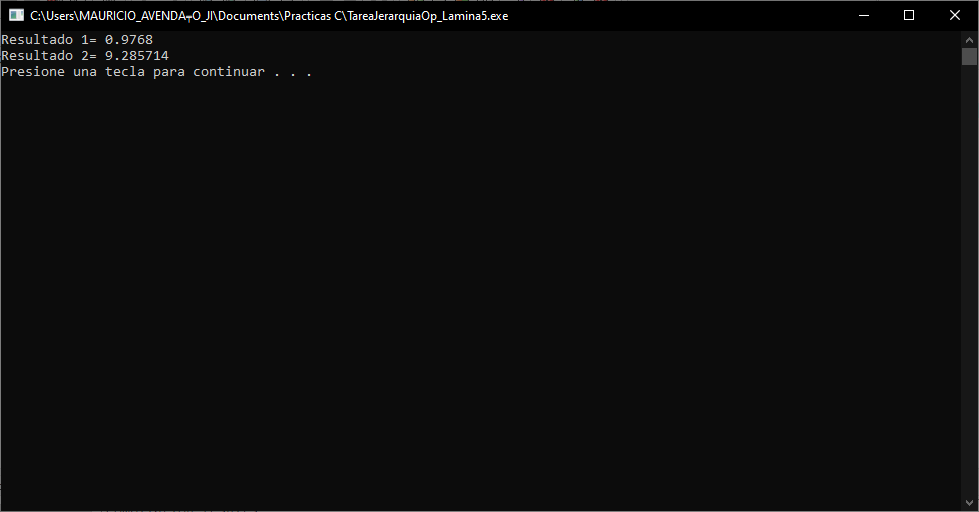
Impresiones en pantalla, y la asignación de los resultados a cada especificador de conversión (%f).

Valor que devuelve el método principal

Función que nos permite detener la ejecución del programa y visualizar los resultados en pantalla



Nota: Cada especificador de acceso se define de acuerdo al tipo de dato que va a recibir, cuando se encuentra un numero entre el, está indicando cuantos números mostrara. Por ejemplo:



Secuencia de escape, salto de línea.



Mensaje

Especificador de conversión de tipo flotante, los números encerrados indican que se van a mostrar 4 números después del punto.

Variable, su valor se asignará en el especificador de conversión y se mostrará en pantalla con ayuda del print formated.

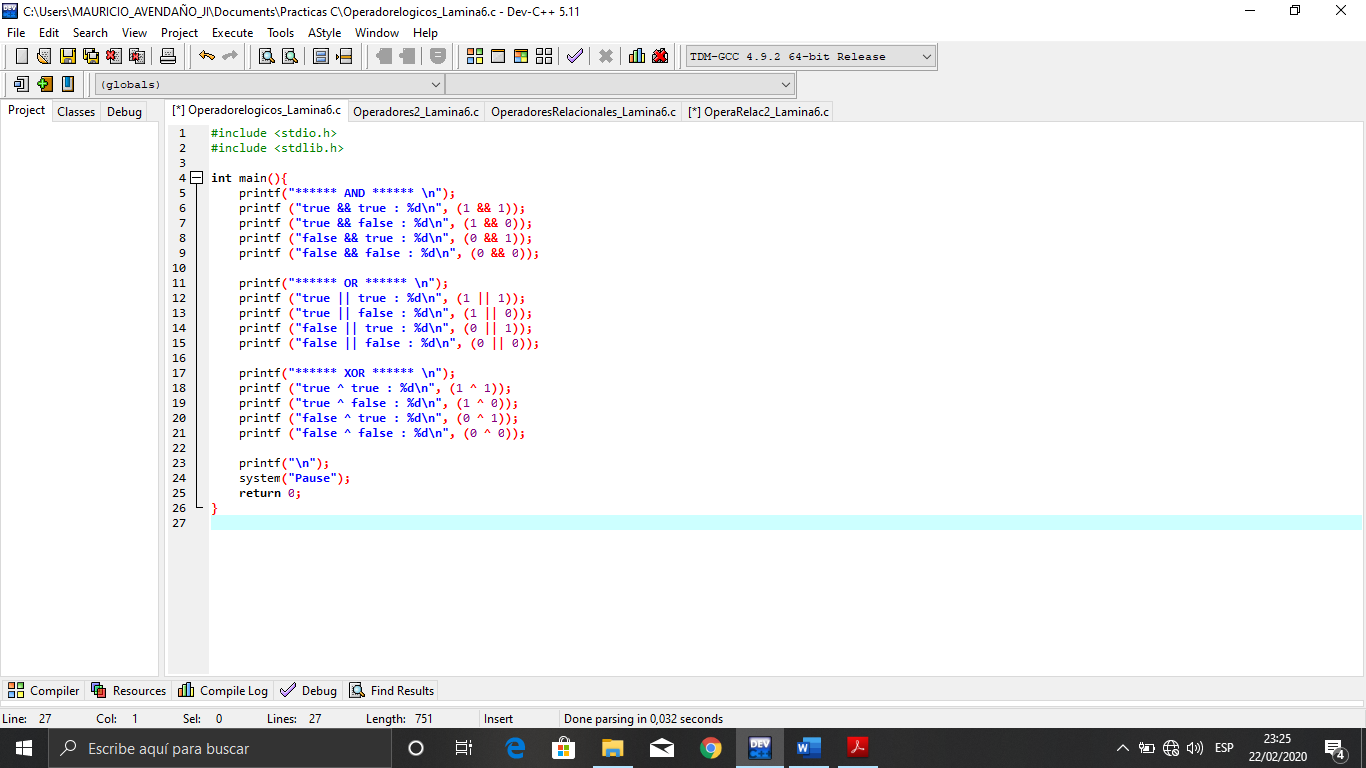
%d🡺 Especificador de conversión, esto depende del tipo de dato que se vaya a recibir.

\n🡺 Secuencia de escape

&&, ||, ^ 🡺Operadores lógicos

**Documento 6**

Ejemplo. Operadores lógicos



Método principal

En la primera línea se manda un mensaje “OR” y en las restantes se hacen las mismas operaciones que en la primera, pero con el operador lógico OR, para que el resultado sea true por lo menos un valor debe ser true.

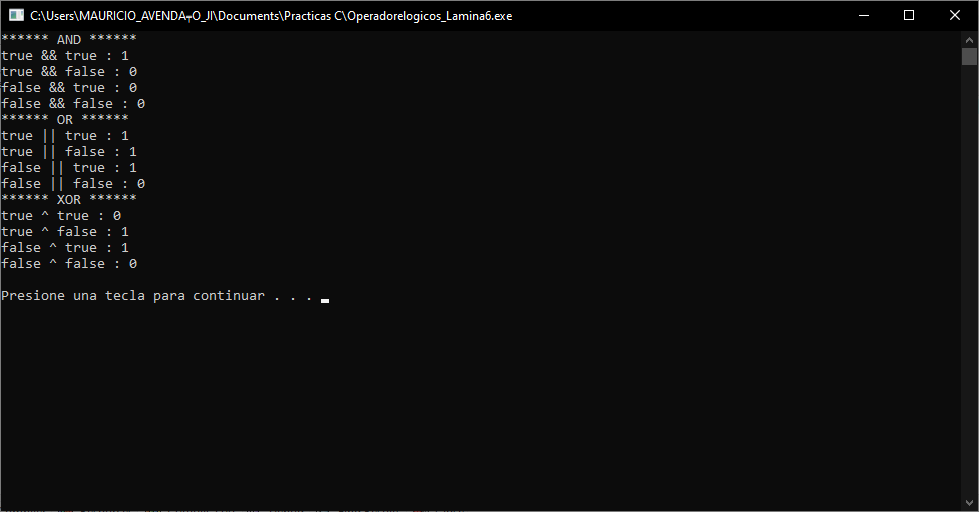
En estas instrucciones es similar a las anteriores, puesto que se siguen utilizando operaciones booleanas, pero en vez de usar OR o AND, se usa XOR, y en este, los valores deben ser true o false, nunca todos true o todos false.

En la primera línea se manda un mensaje “AND” y en las restantes se hacen las operaciones booleanas, es decir, operan con true y false, dependiendo del tipo de operador lógico, en este caso es AND, para que el resultado sea true todos los valores deben ser true.

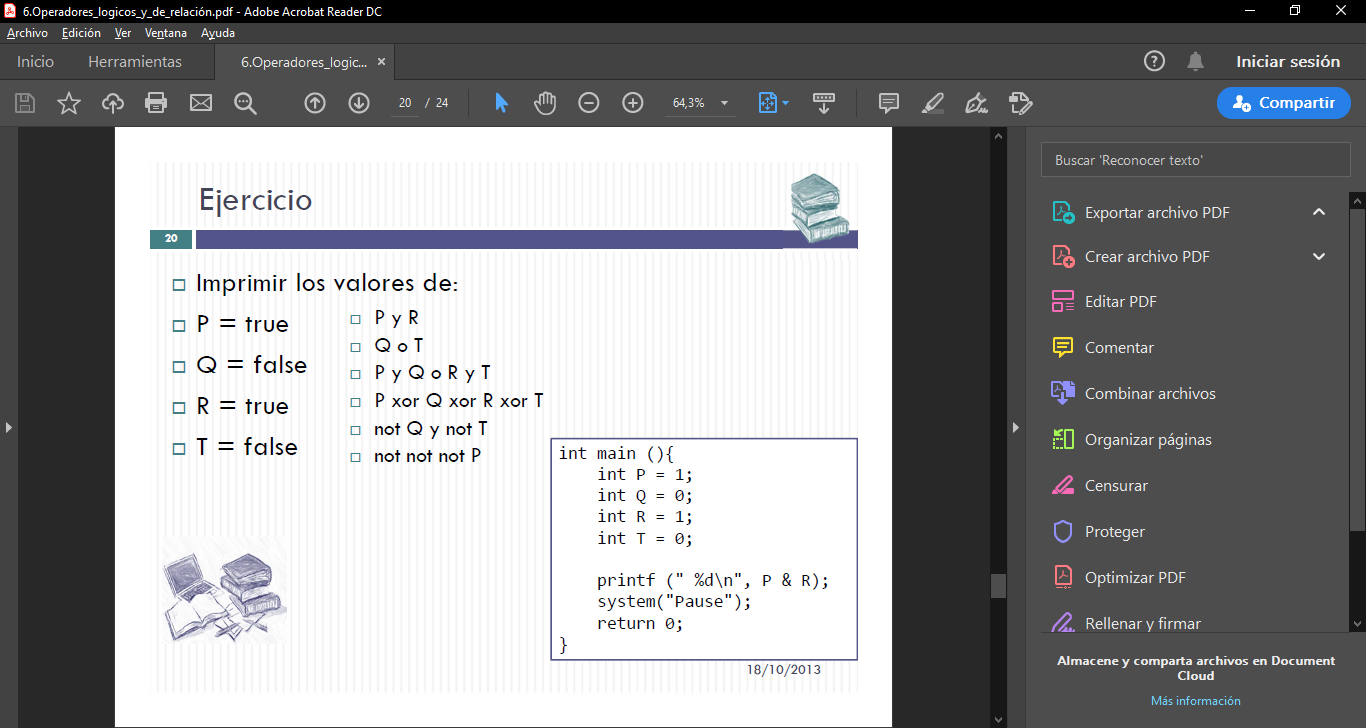
Función que nos permite detener la ejecución del programa y visualizar los resultados en pantalla

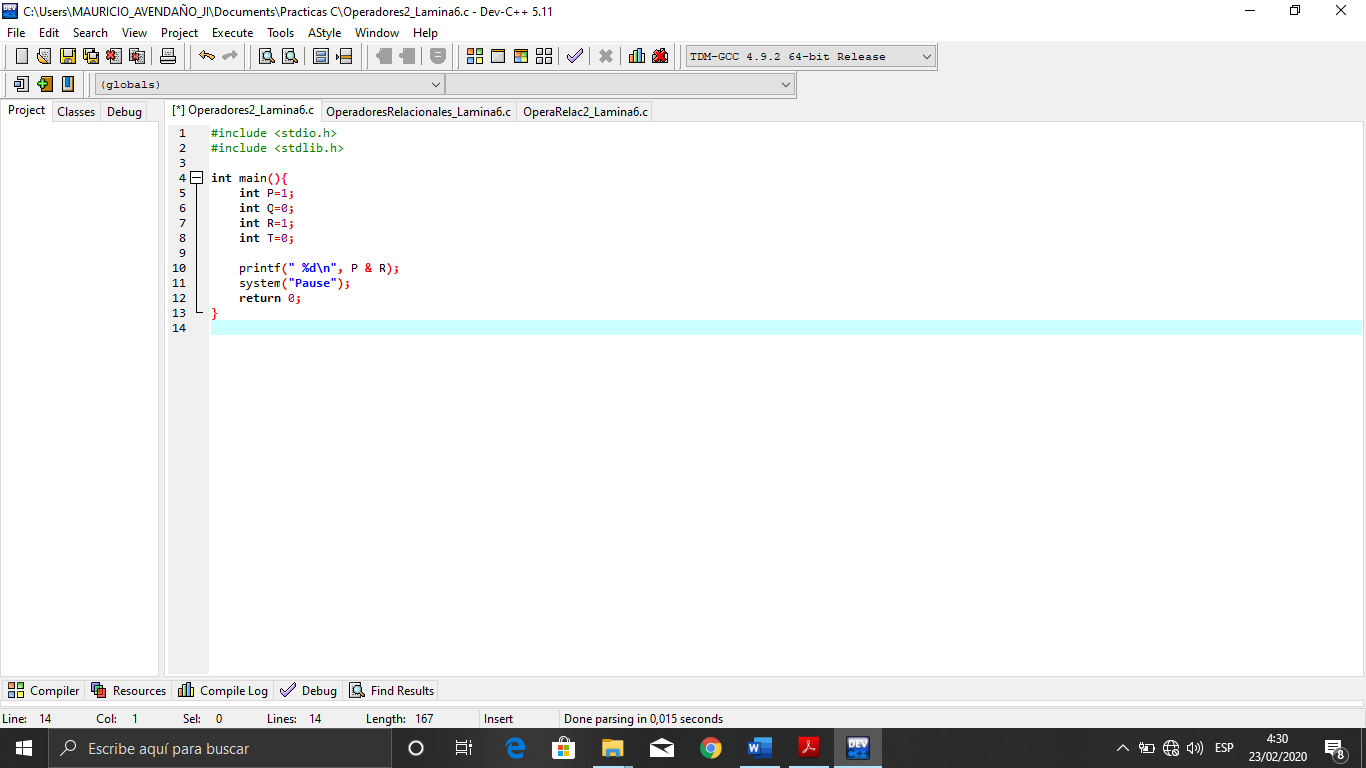
Bibliotecas

Valor que devuelve el método principal



Ejercicio. Operadores lógicos. Imprimir los valores de:





En esta línea, imprimimos la resultante de la operación P&R, y se asigna este valor al especificador de conversión que espera un tipo de dato entero y se imprime en pantalla. P&R=1&1=1

Método principal

Función que nos permite detener la ejecución del programa y visualizar los resultados en pantalla

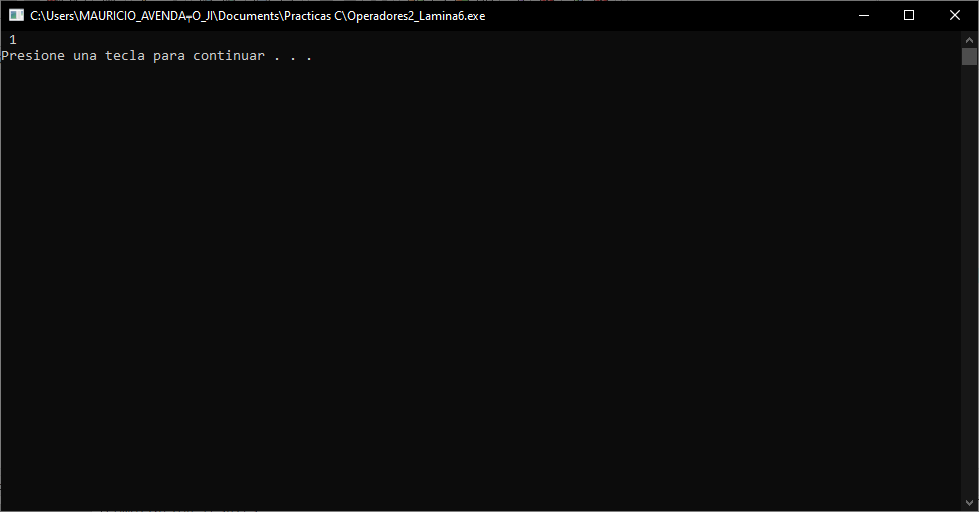
Declaración de variables, de tipo entero, pero, aunque estos contienen valores enteros, se consideran como booleanos, 1 corresponde a true y 0 a false.

Bibliotecas

&d 🡺especificador de conversión

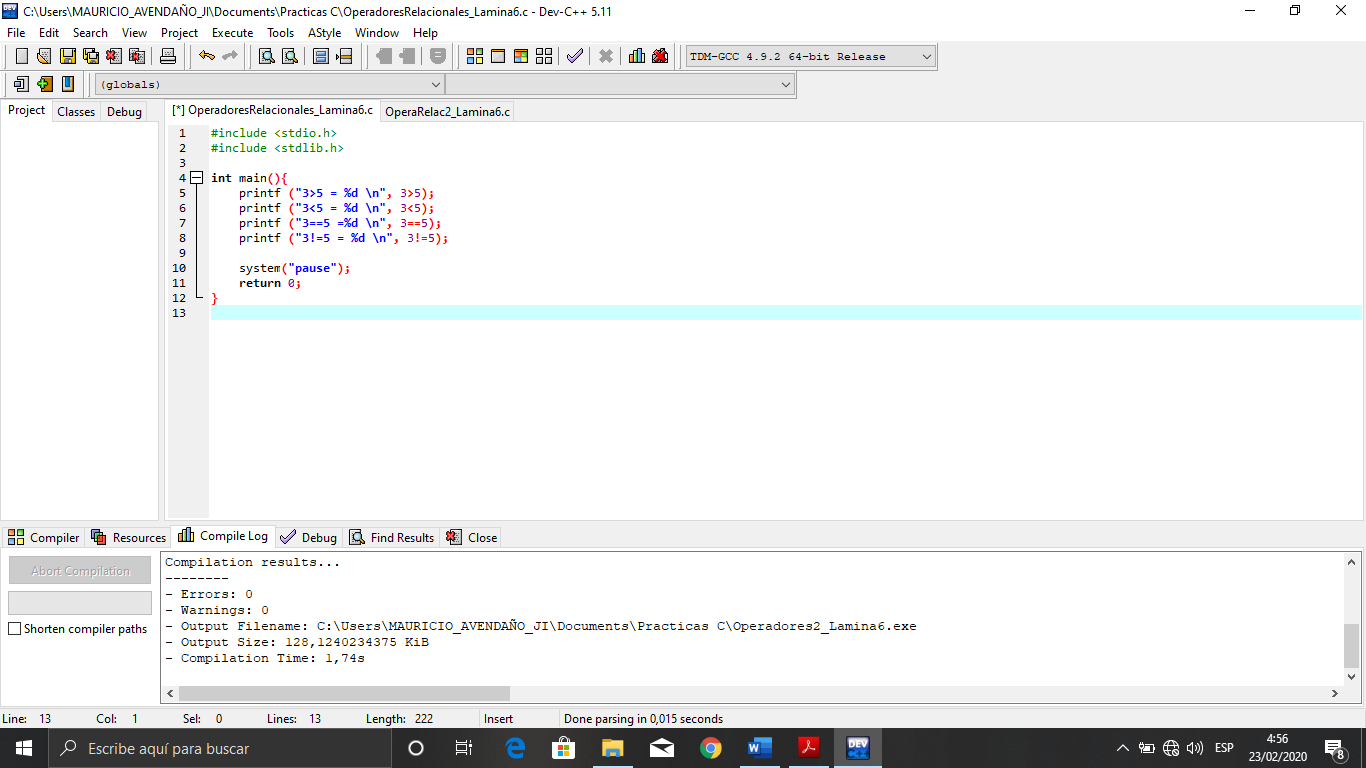
\n 🡺 Secuencia de escape, salto de línea

Valor que devuelve el método principal



Ejemplo. Operadores relacionales.

\n 🡺 Secuencia de escape, salto de línea



Operaciones relacionales, también conocidas como condiciones.

Especificador de conversión

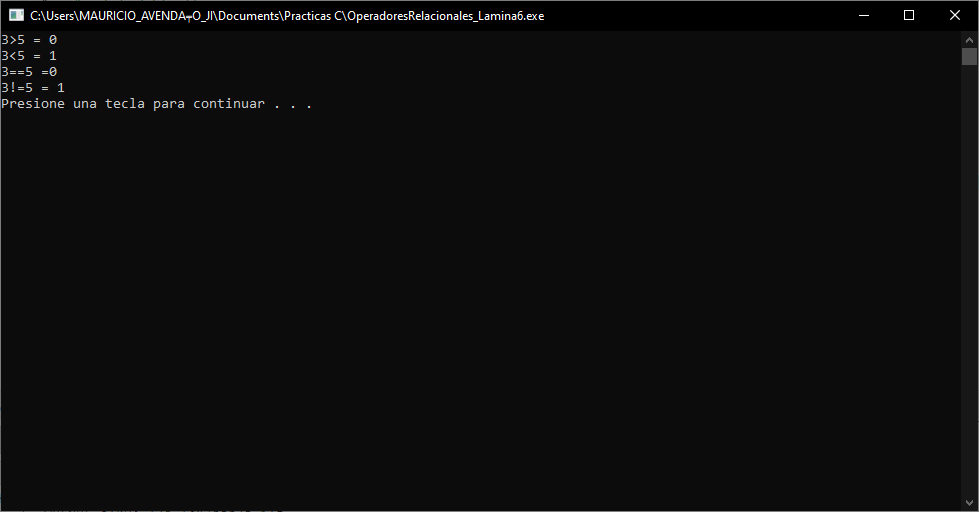
Estas instrucciones, consisten en comparar números, el resultante, que es un valor booleano, se asignara en el especificador de conversión de tipo entero, pero como ya habíamos mencionado antes, el 1 y 0 son valores enteros pero también son considerados valores booleanos, dependiendo del uso que les den.

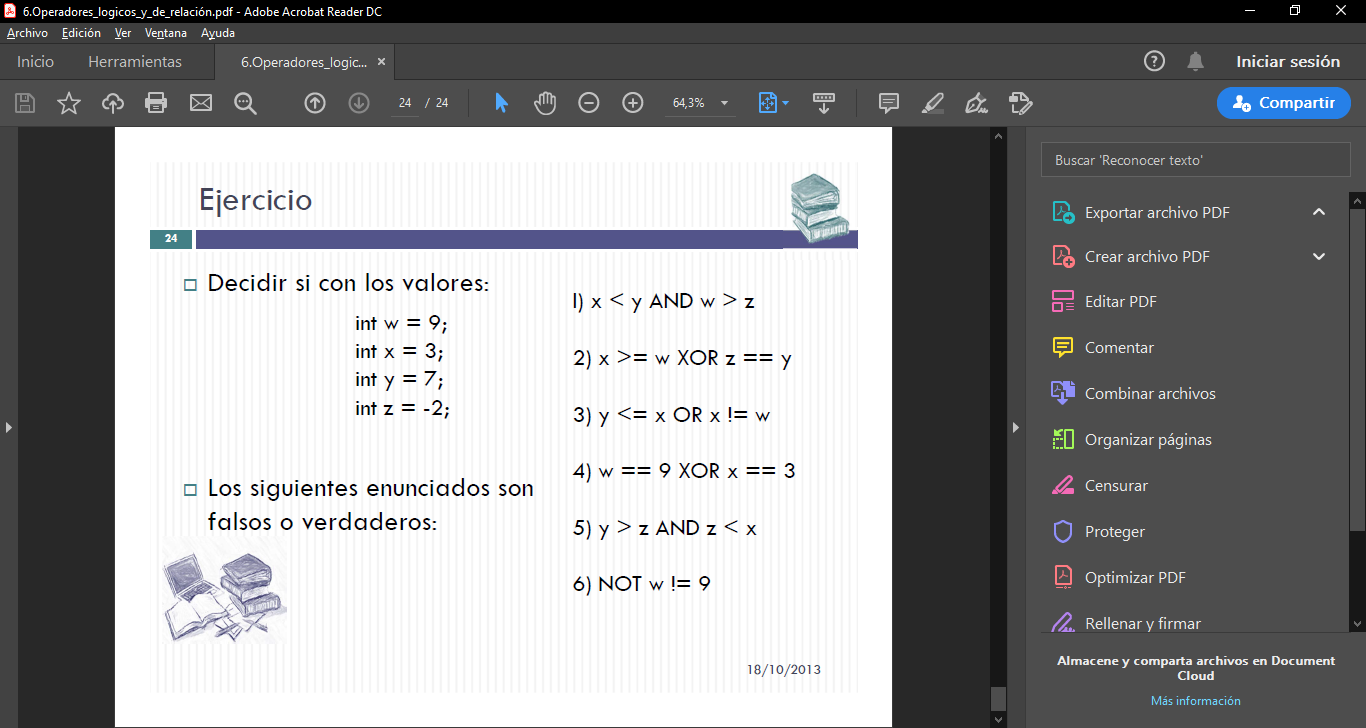
Función que nos permite detener la ejecución del programa y visualizar los resultados en pantalla

Método principal

Bibliotecas

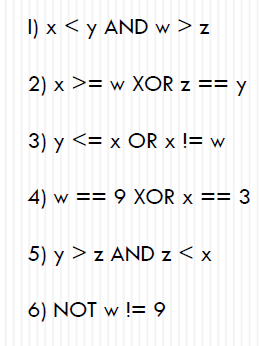
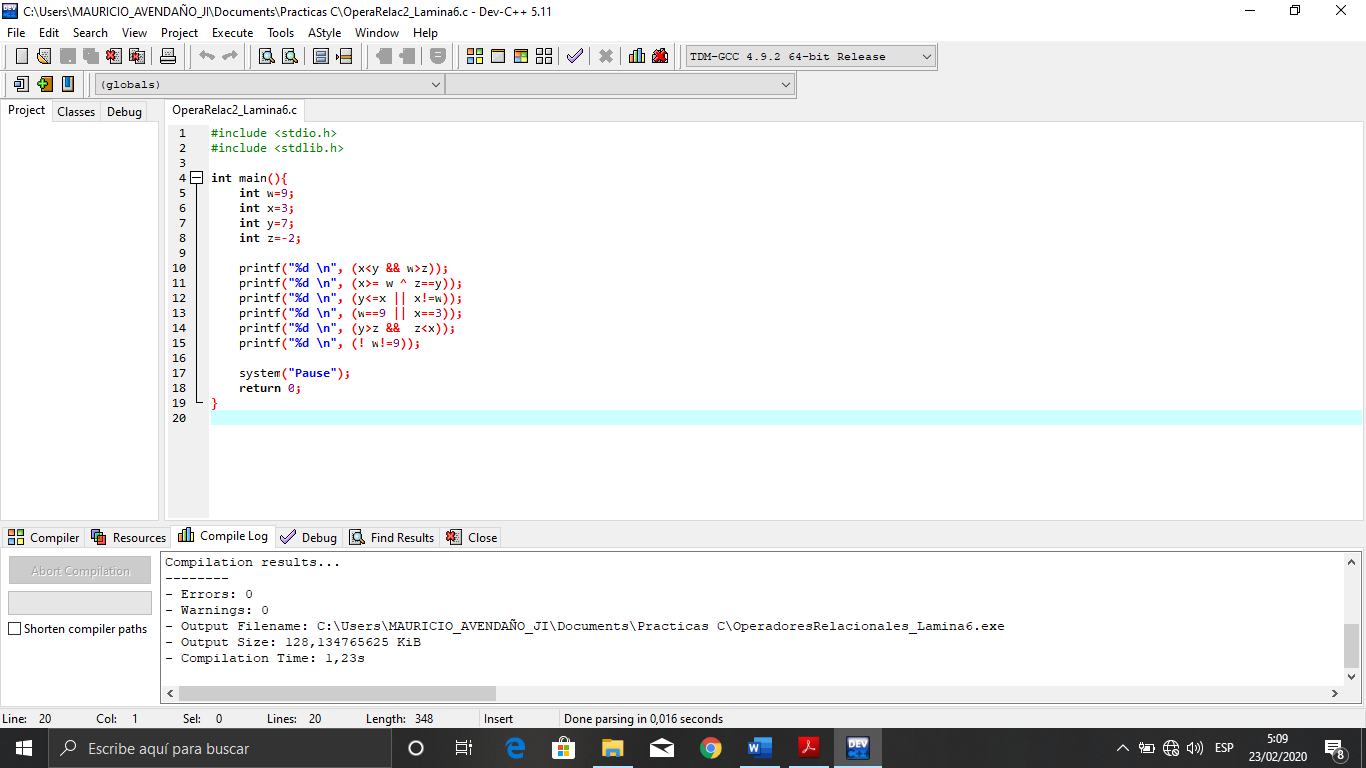
Valor que devuelve el método principal





Ejercicio. Operadores lógicos. Decidir si con los siguientes valores:

Los siguientes enunciados son verdaderos o falsos.



Método principal

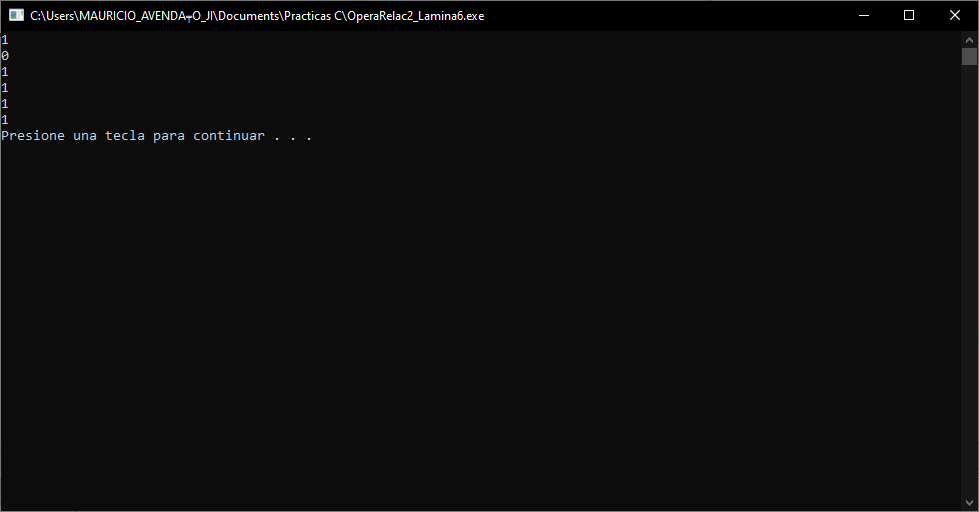
Bibliotecas

En estas líneas se ejecutan las operaciones relacionales con operadores lógicos, primero se hacen las relacionales y luego las lógicas, el resultado de cada una se asigna al especificador de conversión, y se imprime en pantalla con un salto de línea al finalizar. Las operaciones se hacen de acuerdo a los valores que tienen las variables.

Declaración de variables, de tipo entero, cada una con su respectivo valor.

Valor que devuelve el método principal

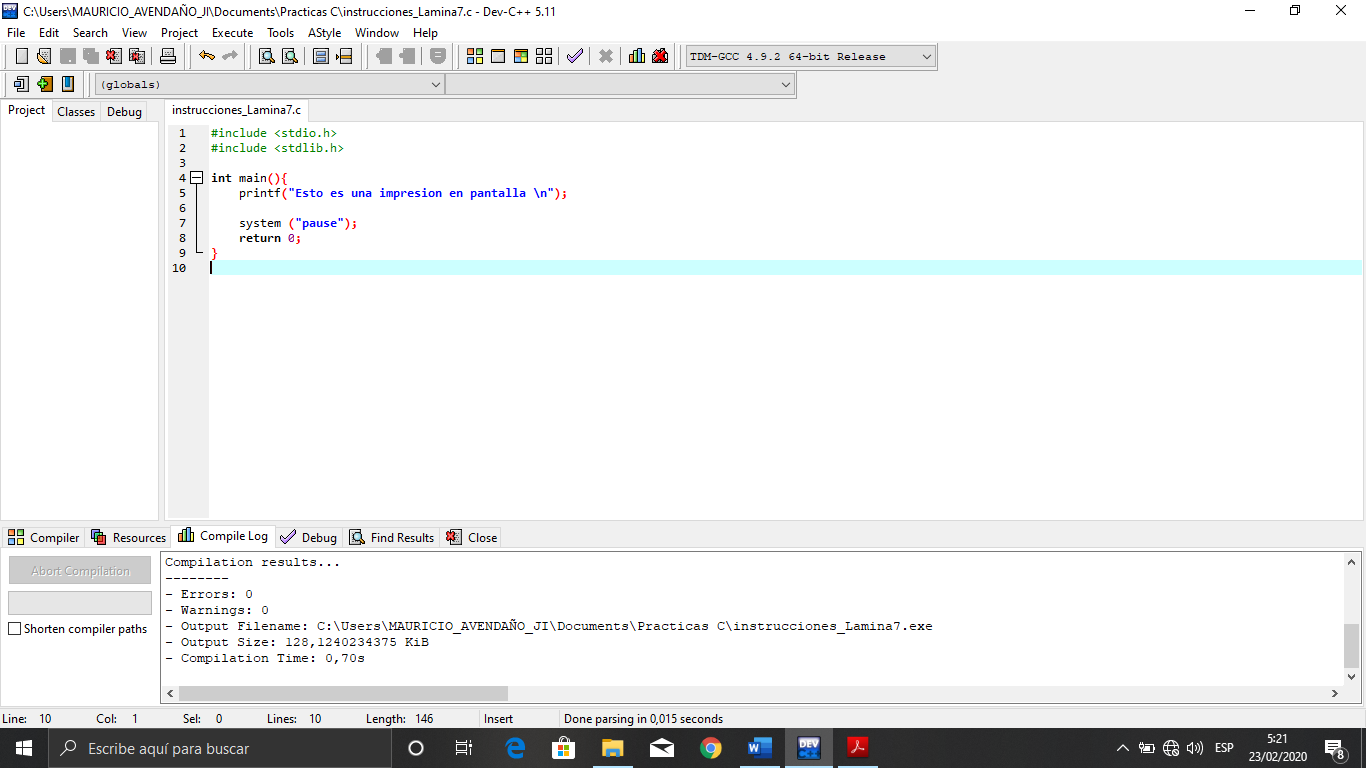
Función que nos permite detener la ejecución del programa y visualizar los resultados en pantalla



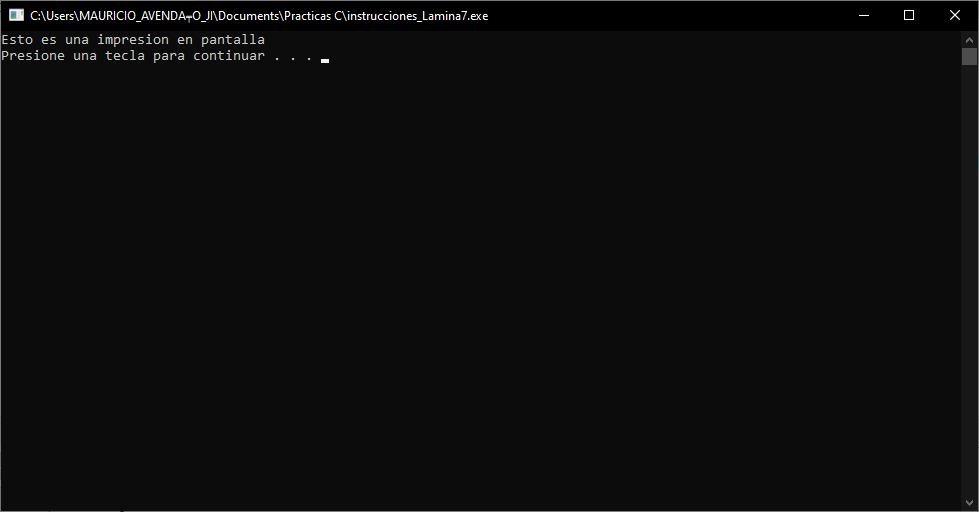
**Documento 7**

printf()

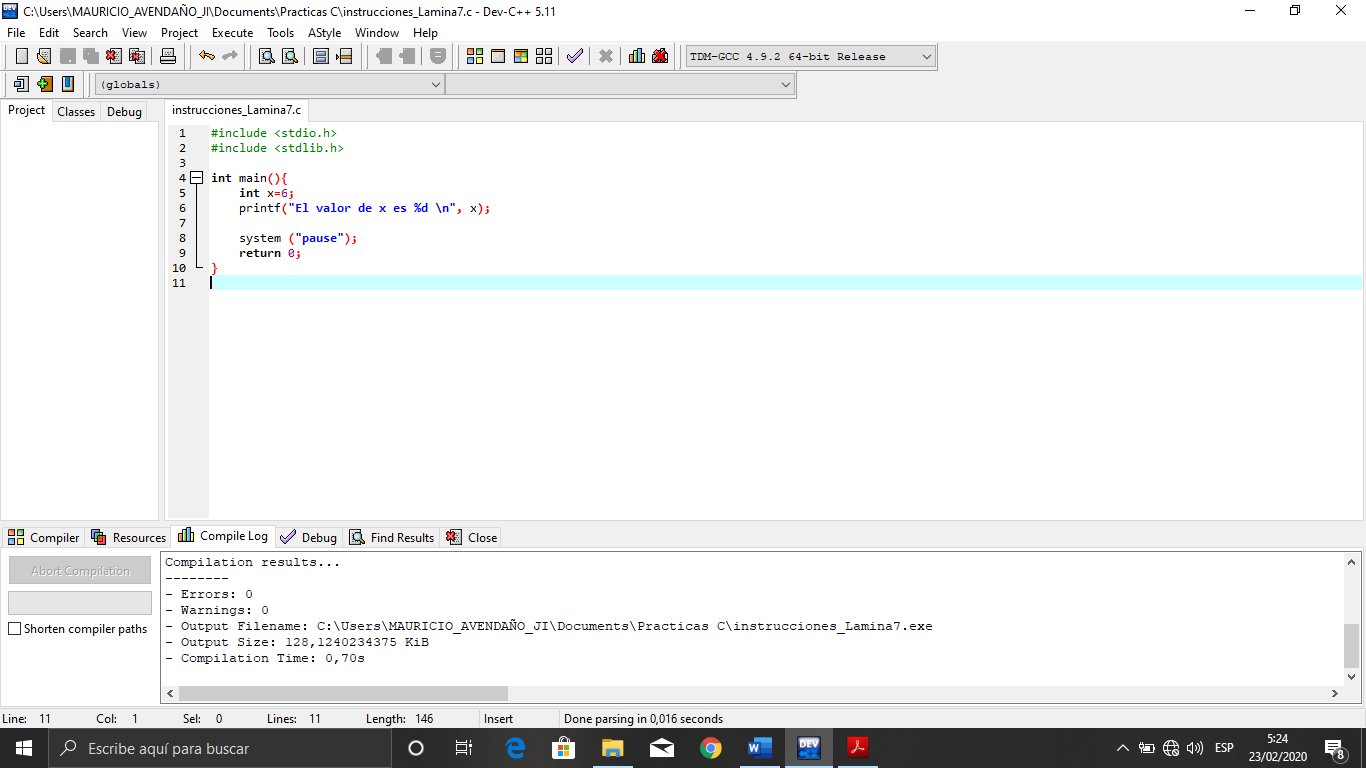
* Imprimir un mensaje simple en pantalla



Secuencia de escape



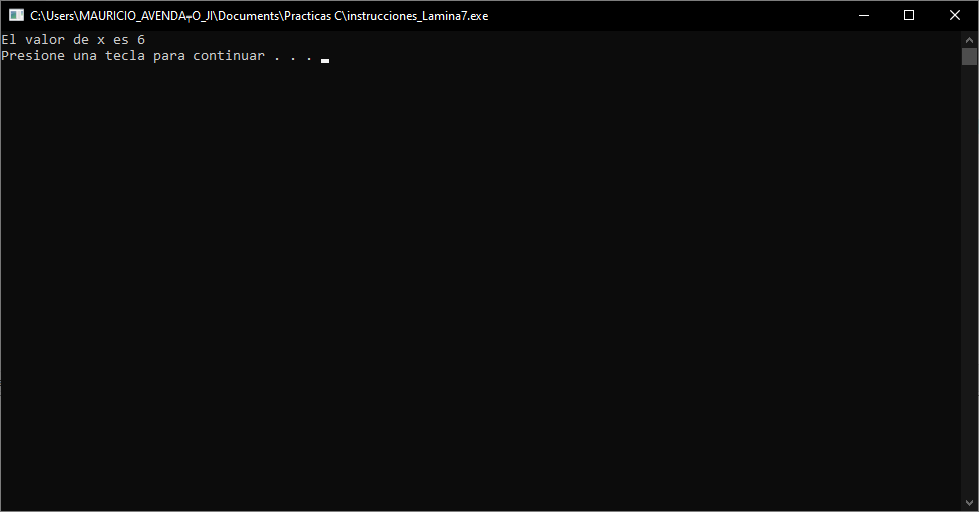
* Imprimir mensajes y valores de variables



Especificador de conversión

Secuencia de escape

Variable



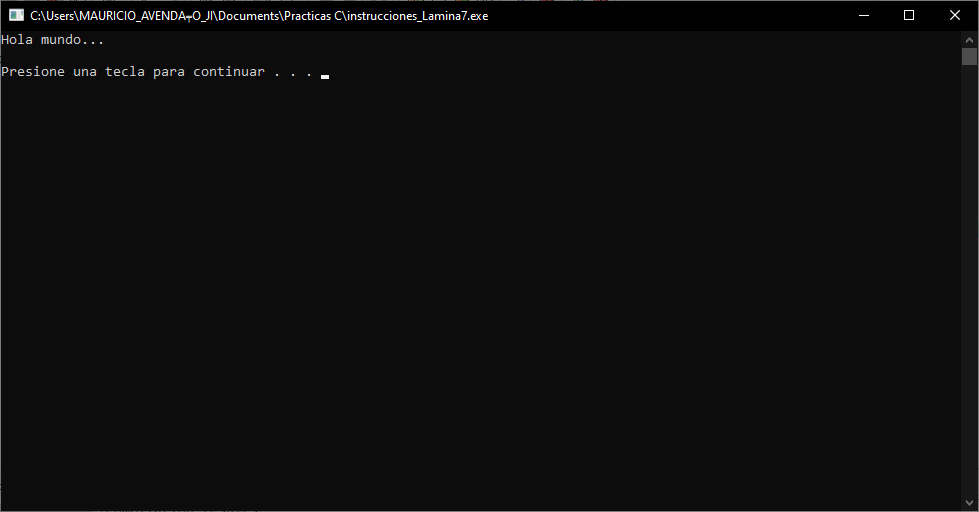
puts()

Imprime mensajes en pantalla, pero no despliega variables, debe incluir el encabezado stdio.h

Encabezado

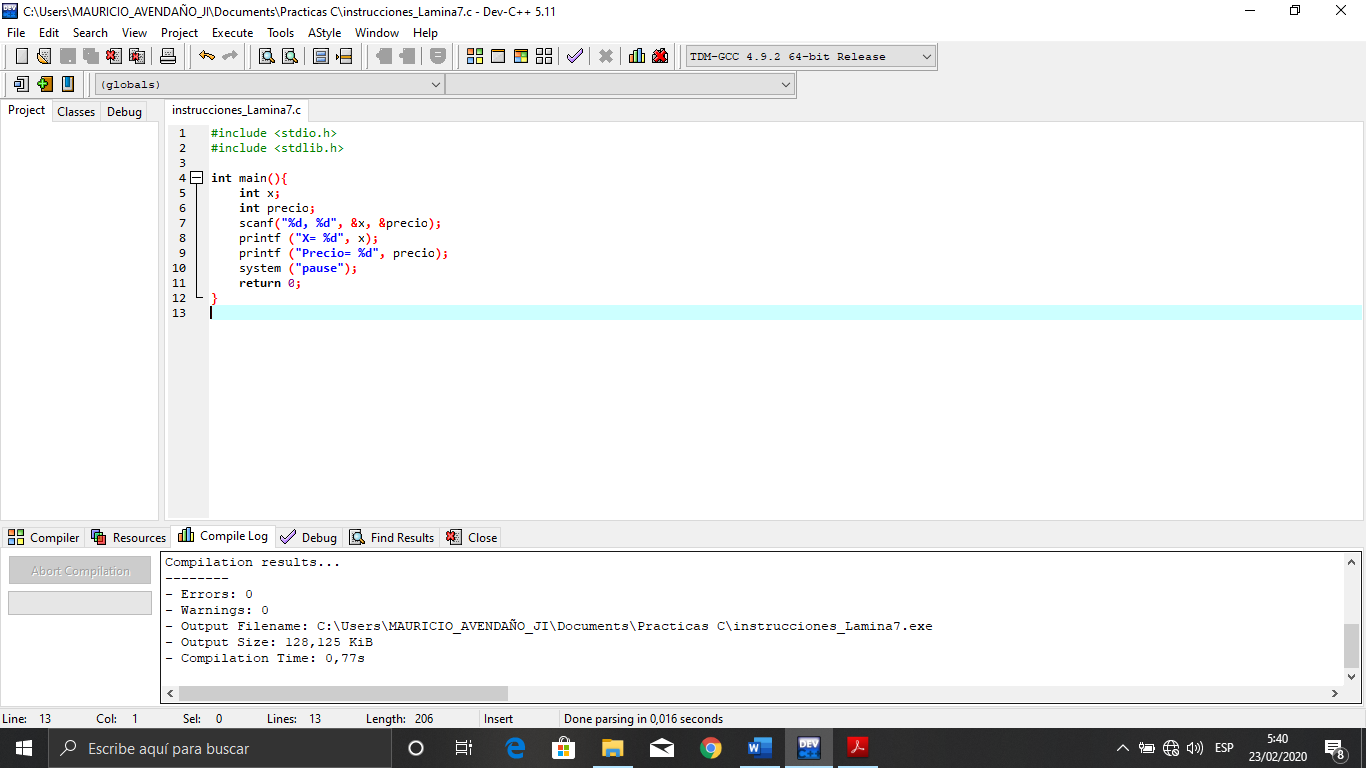


Secuencia de escape



scanf()

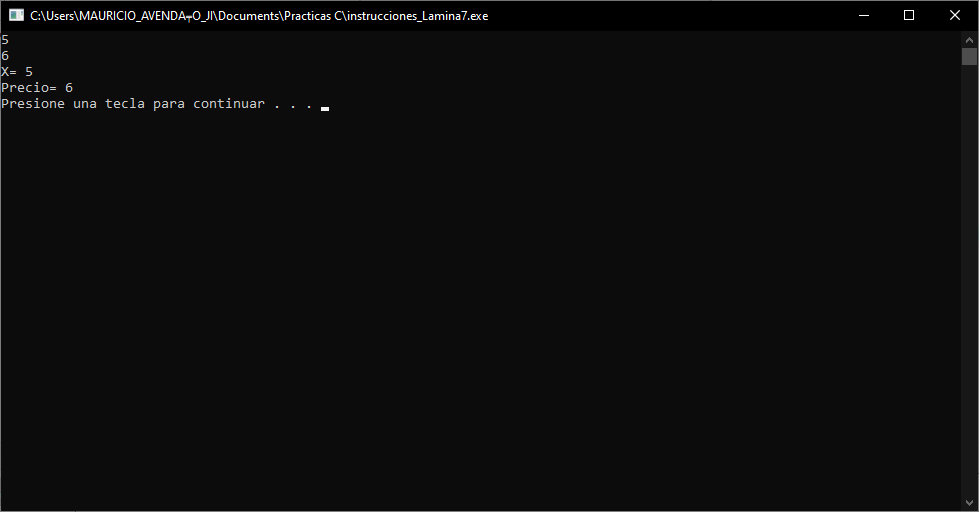
Lee datos del teclado y asigna los datos de entrada a una o mas variables de dicho programa, pero ahora este utiliza un operador de dirección.



Nota: se hizo uso de printf() para demostrar que en efecto los datos de entrada se asignan a variables del programa. Como se puede observar en la parte inferior, la asignación fue la correcta.

Operador de dirección

Datos de entrada, es decir, los datos que ingresamos por teclado.

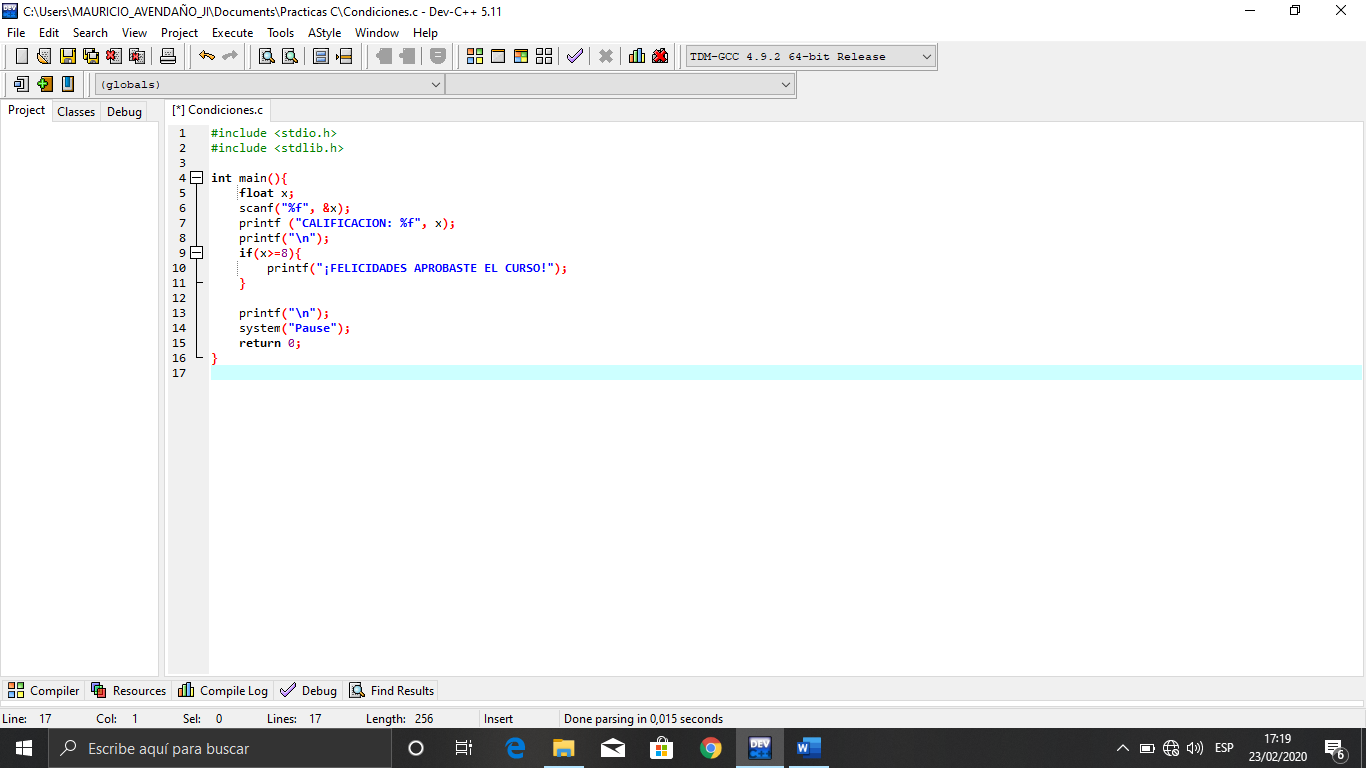


Asignaciones de los datos de entrada a las variables.

**Archivos extra**

Esta función permite ingresar datos por teclado, consta del especificador de conversión, seguido del operador de dirección y el nombre de la variable en la que se alojara el dato ingresado.

Bibliotecas



Función que nos permite detener la ejecución del programa y visualizar los resultados en pantalla

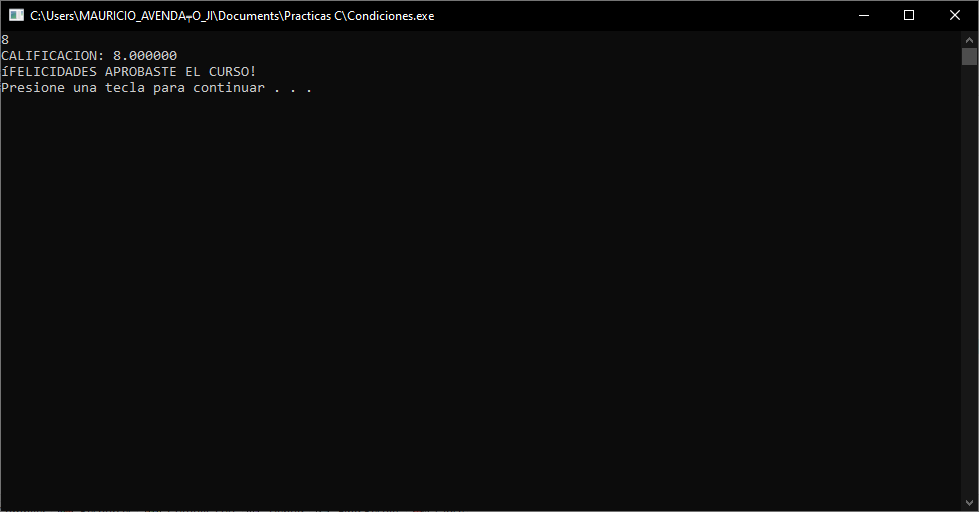
Mensaje que manda la calificación del estudiante, la variable x ya tiene el valor de entrada, se asigna al especificador y se imprime en pantalla, después de esa línea se hace la comparación.

Método principal

Condición, si el valor de x es mayor o igual a 8 va a imprimir el mensaje de: felicidades, aprobaste el curso, pero si no se cumple la condición simplemente acaba la ejecución del programa.

Declaración de variable, de tipo flotante llamada x.

Valor que devuelve el método principal

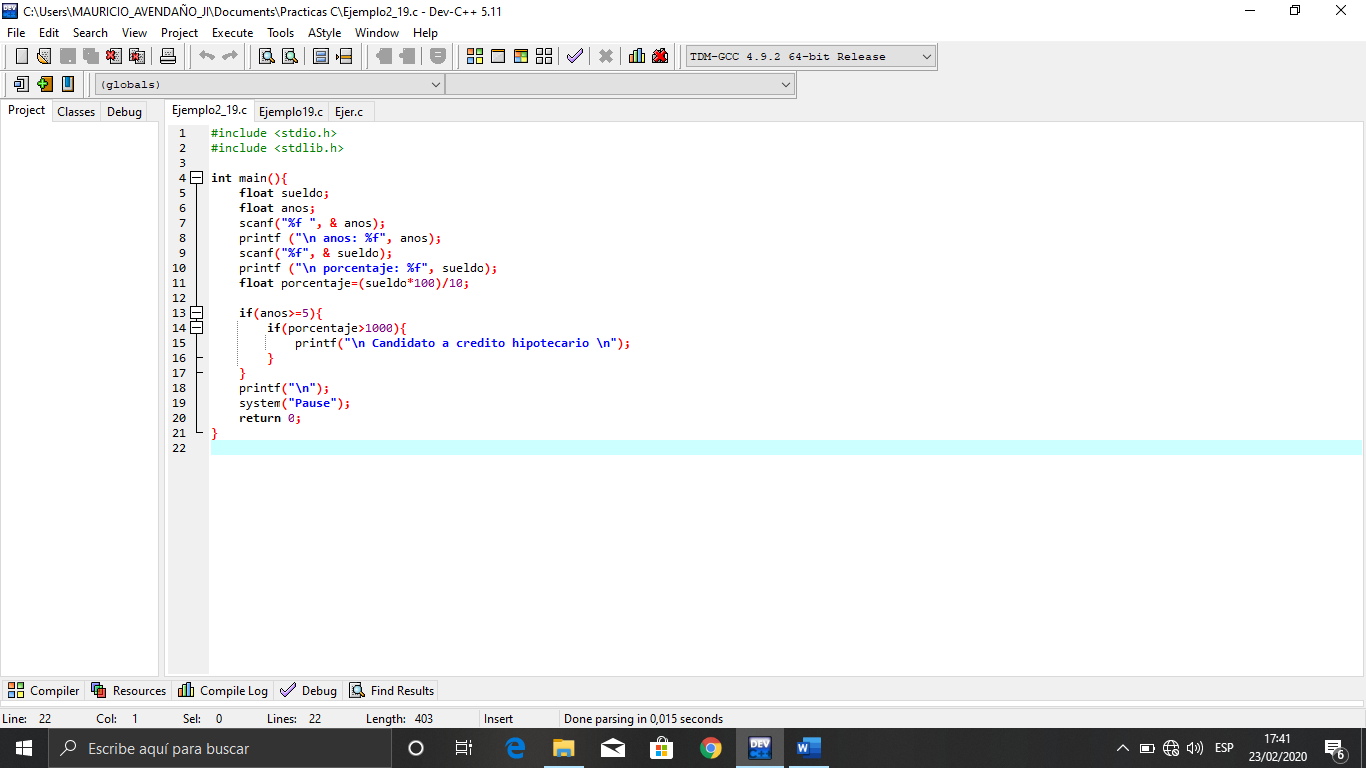


Dato ingresado por teclado, como se puede observar la condición si se cumple porque no es mayor a 8 pero si es igual, así que entra a la condición e imprime el mensaje de aprobado.

Mensaje que se encuentra dentro de la condición y en el caso de que la condición resulte verdadera se ejecutara, en caso contrario saltara esa línea y continuara. En este caso si fue verdadera por lo que si se ejecutó esa instrucción.

Mensaje de la calificación.

Esta función permite ingresar datos por teclado, consta del especificador de conversión, seguido del operador de dirección y el nombre de la variable en la que se alojara el dato ingresado.



Primero se hacen las operaciones para determinar el 10% a comparar, el resultado se guardará en la variable porcentaje, después pasa a la condición, si los anos ingresados son mayores o igual a 5 y e porcentaje más de 1000, es candidato a obtener el crédito, pero si no se cumplen ambas no puede.

Función que nos permite detener la ejecución del programa y visualizar los resultados en pantalla.

Mensaje, consta del especificador de conversión, en el que se alojará el valor de la variable y se mostrará en pantalla, el especificador aquí estará esperando un valor de tipo float.

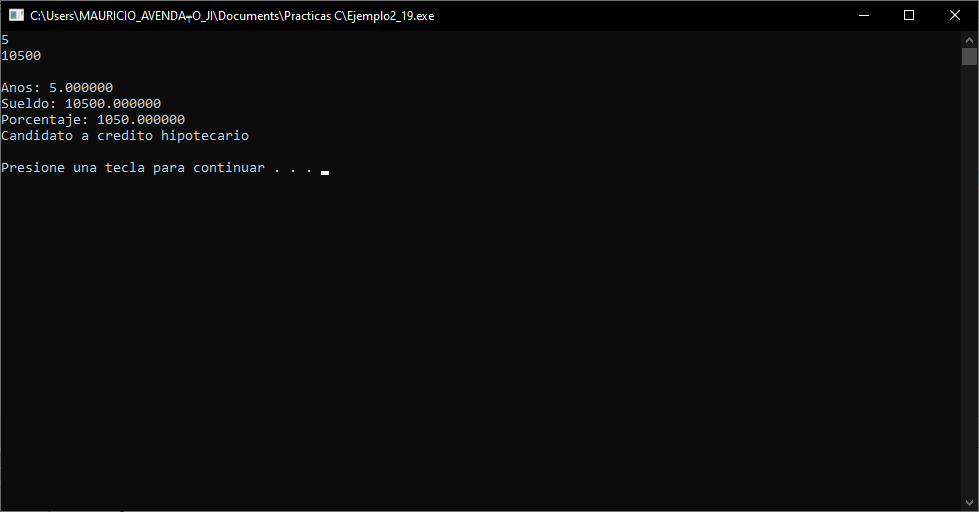
Declaración de variables, de tipo flotante llamada sueldo y anos.

Valor que devuelve el método principal

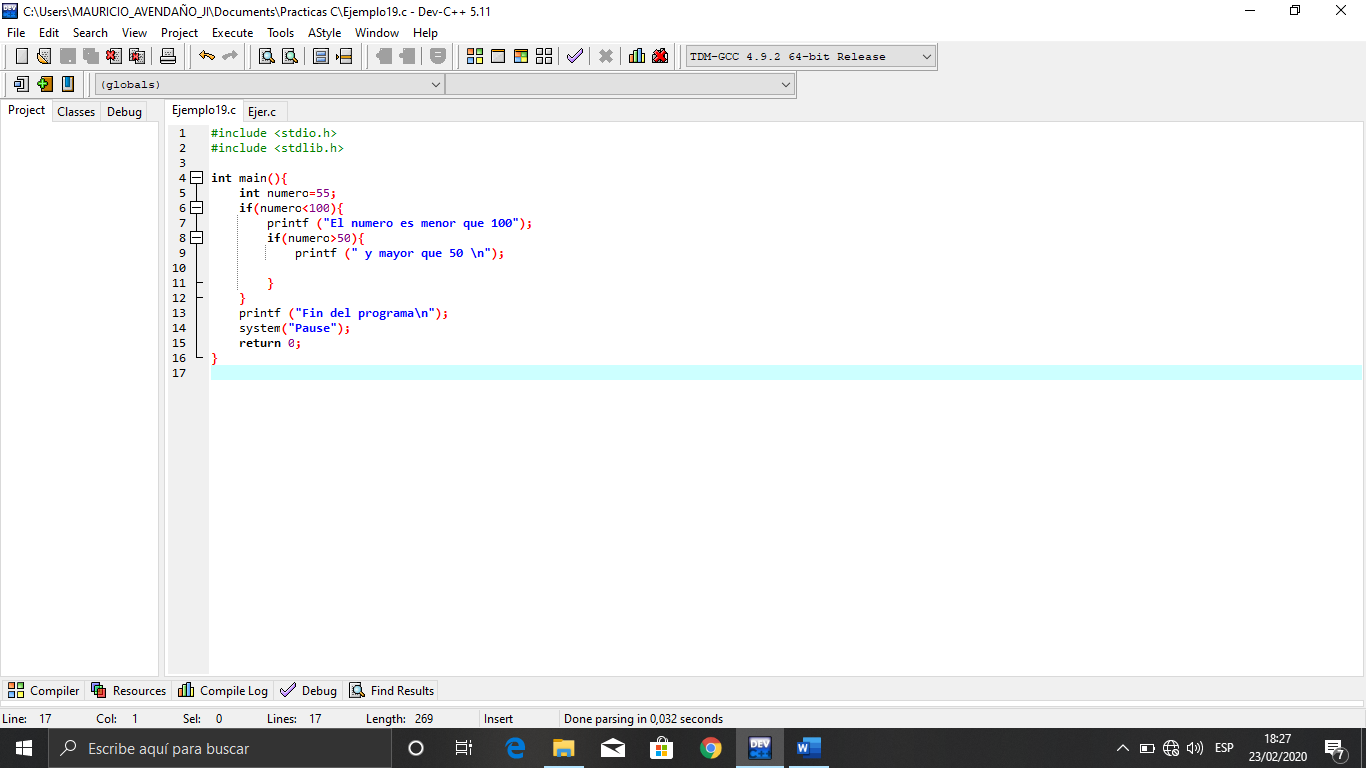
Método principal

Bibliotecas

Datos ingresados por teclado, como nos podemos dar cuenta ingreso 5 años y su sueldo de 10500, el 10% de su sueldo es de 1050 y como la condición pide que por lo menos tenga 5 años y que el 10% de su sueldo sea mayor de 1000, la persona si cumple con los requisitos para pedir el crédito hipotecario, por lo que si se lo otorgaran, pero en caso que no cumpliera con las dos o ninguna, no se le otorgaría.



Mensaje que se encuentra dentro de la condición y en el caso de que la condición resulte verdadera se ejecutara, en caso contrario saltara esa línea y continuara. En este caso si fue verdadera por lo que si se ejecutó esa instrucción.



Condición que evalúa si número es menor a 100, si es así, se imprime el mensaje: el número es menor que 100, y continua evaluando, si el número es mayor que 50, también imprimirá el mensaje: y mayor que 50, pero dado el caso que la primera condición fuera falsa, no imprimiría ni el primer mensaje y mucho menos evaluaría a la segunda condición.

Declaración de variable, de tipo entero llamada número, inicializada en 55.

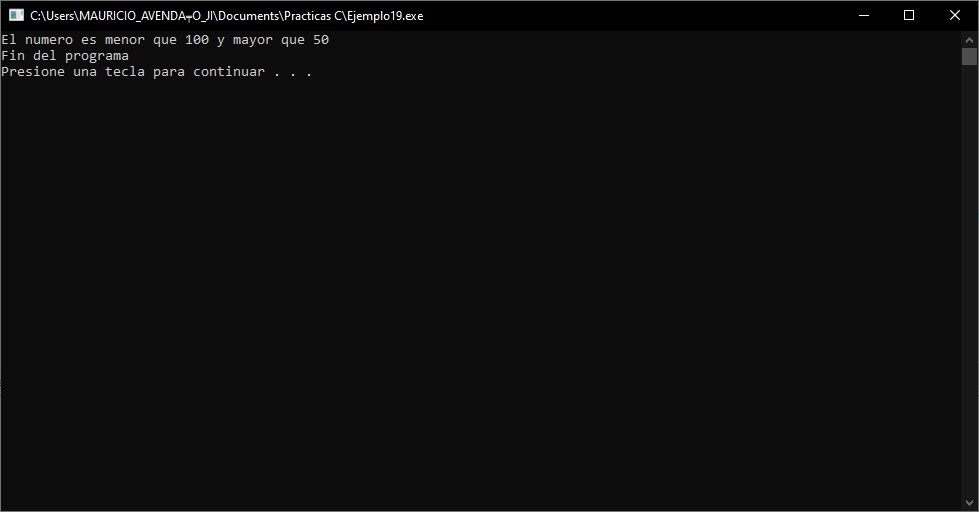
Mensaje y salto de línea al final

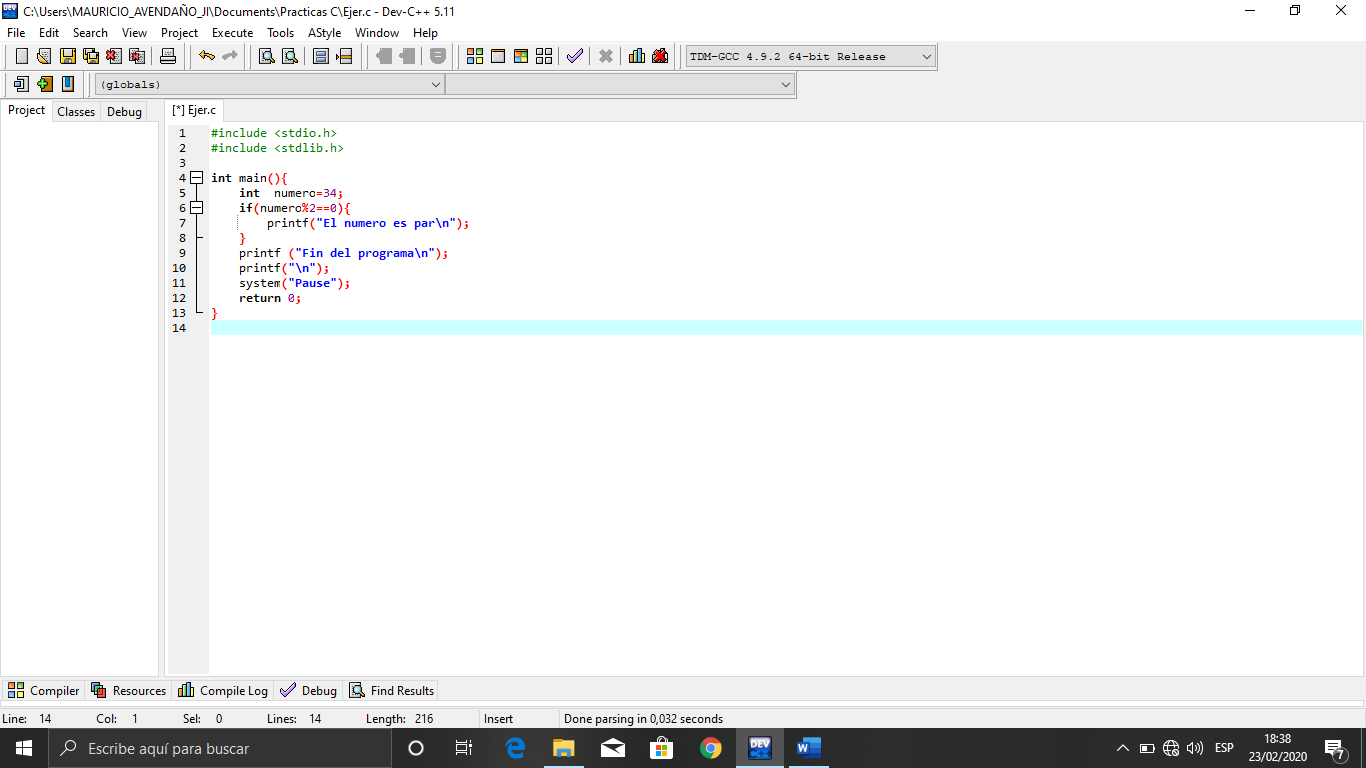
Método principal

Función que nos permite detener la ejecución del programa y visualizar los resultados en pantalla.

Valor que devuelve el método principal

Bibliotecas





Condición que evalúa si el número es par o no, esto se hace obteniendo el residuo del número entre 2, si el residuo es cero quiere decir que es par, pero si es diferente de 0 es impar

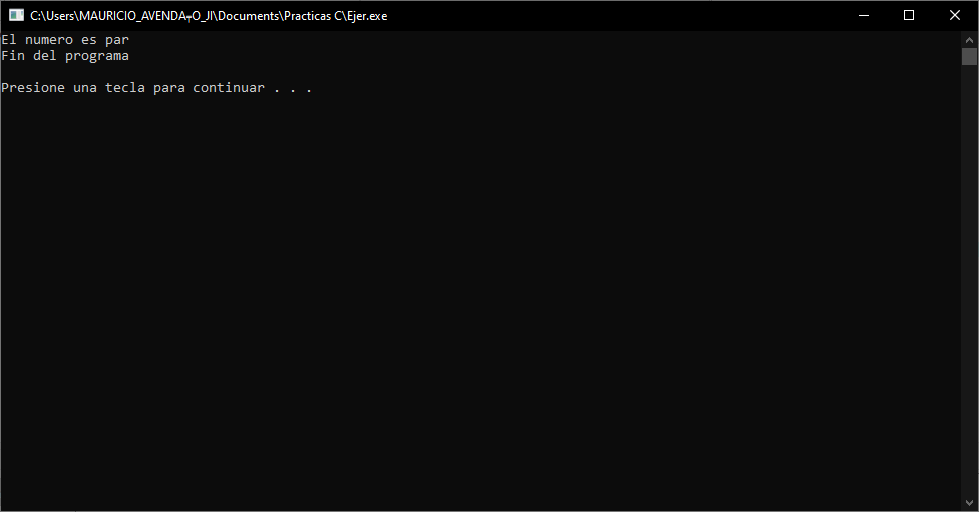
Bibliotecas

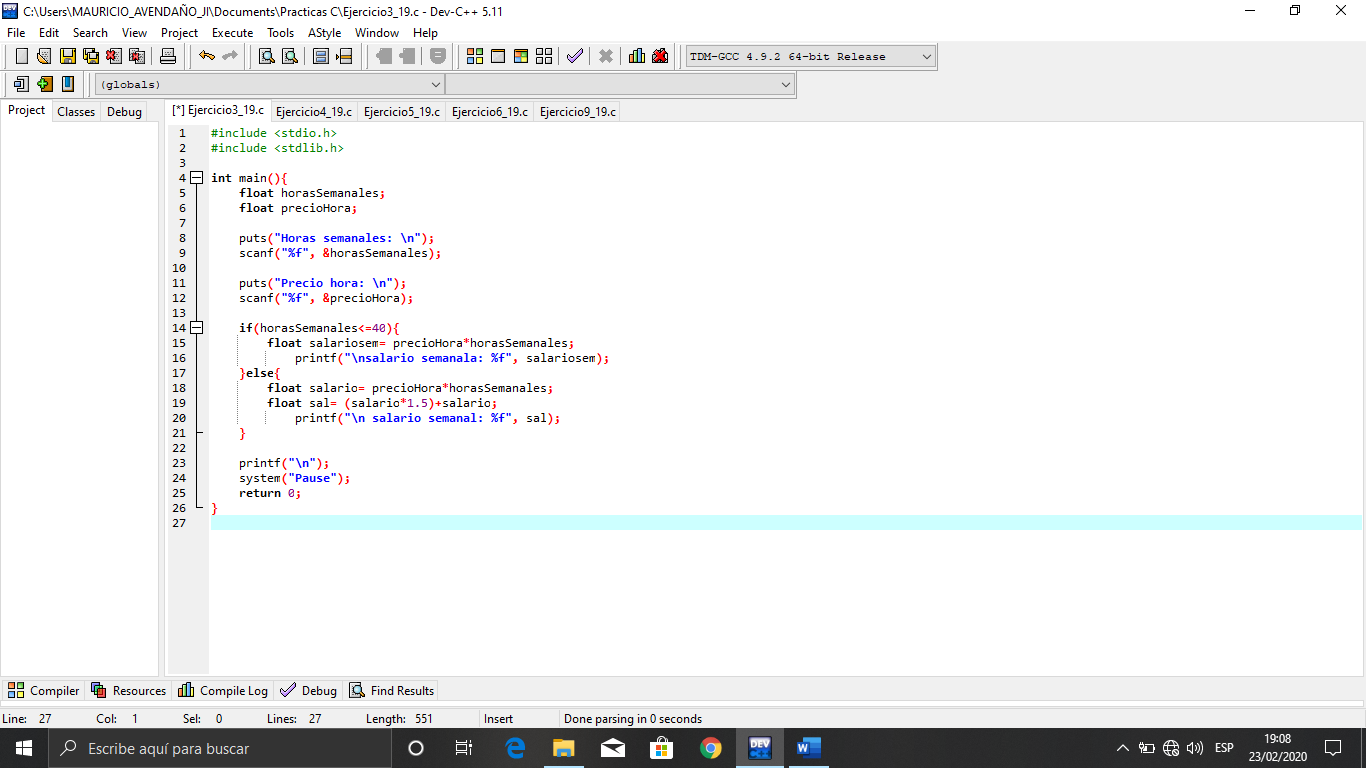
Declaración de variable, de tipo entero llamada número, inicializada en 34.

Método principal

Función que nos permite detener la ejecución del programa y visualizar los resultados en pantalla.

Valor que devuelve el método principal





Función que permite la entrada de datos por teclado, consta de un operador de dirección para la variable y un especificador de conversión del tipo de dato que se espera recibir.

Método principal

Bibliotecas

Función que nos permite detener la ejecución del programa y visualizar los resultados en pantalla.

Puts() permite mandar un mensaje en pantalla.

Operador de dirección

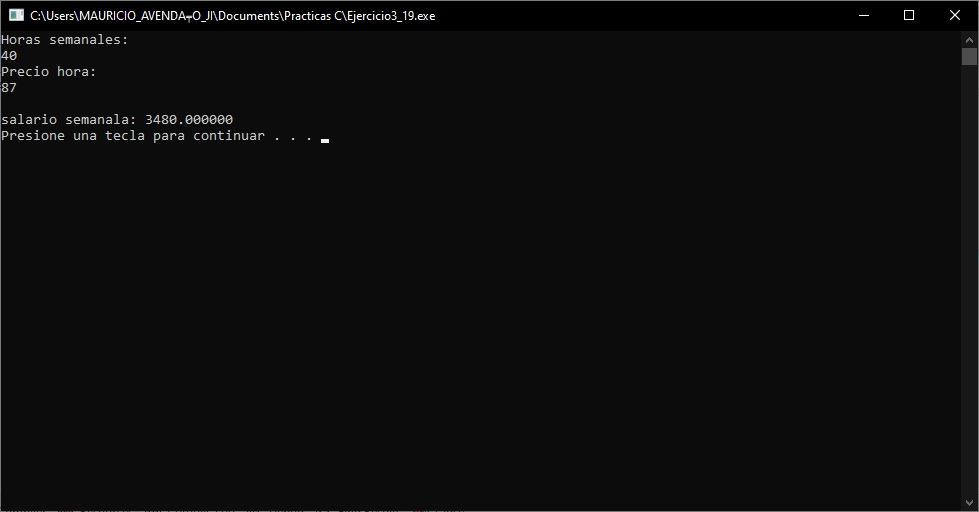
Especificador de conversión

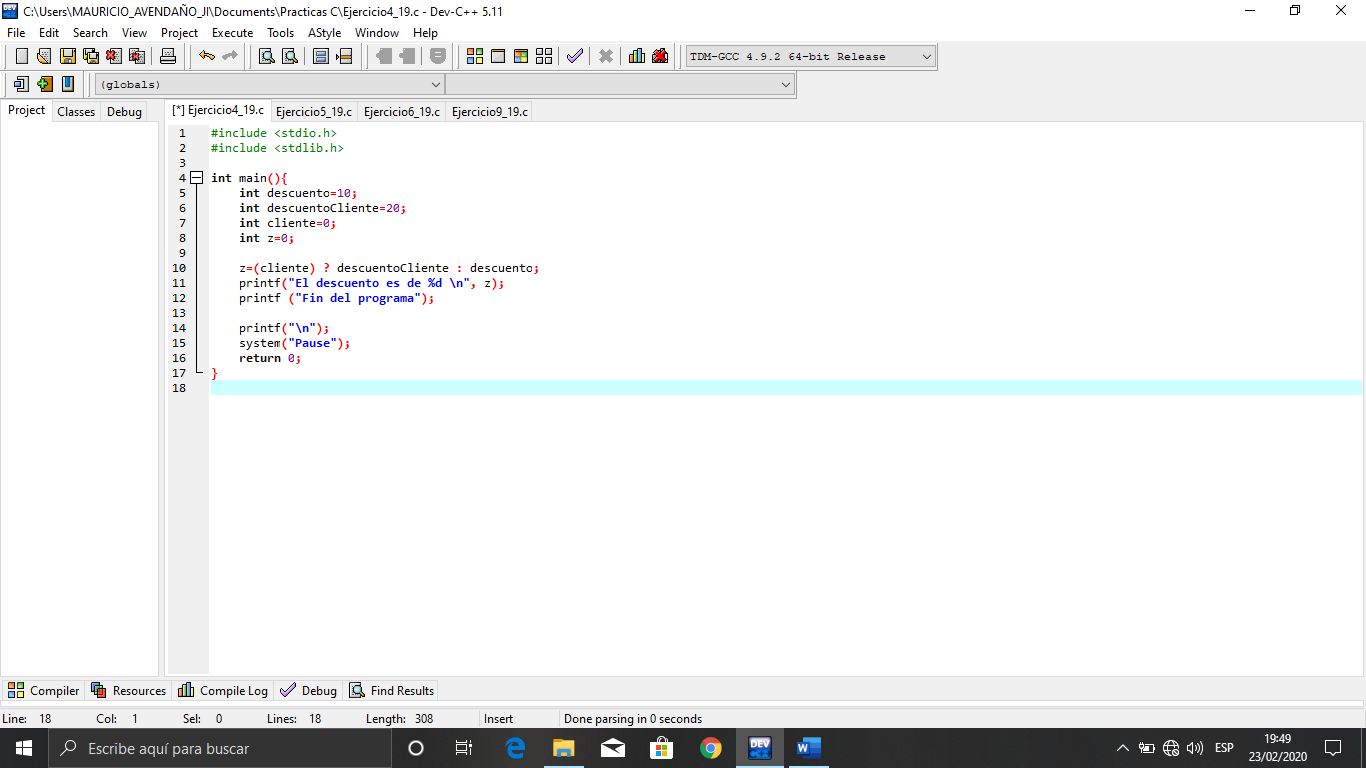
Condición que evalúa las horas semanales, si son menor o igual a 40, entonces se le paga normal, pero si son mas entonces se le paga a razón de 1.5 mas del salario normal.

En las varibles salariosem, salario y sal, se guardan los resultados de las operaciones. Printf() sirve para imprimir en pantalla un texto y valores de variables, consta de un especificador de conversión para asignar el valor de alguna variable que se desee mostrar.

Declaración de variables, de tipo flotante llamadas horasSemanales y precioHora.

Valor que devuelve el método principal





El operador ternario indica que si el primer enunciado es verdadero toma el segundo enunciado, pero si el primero es falso toma en cuenta el tercero, luego con ayuda del printf() mostramos en pantalla el valor que tomo z, y el mensaje de finalizado.

Bibliotecas

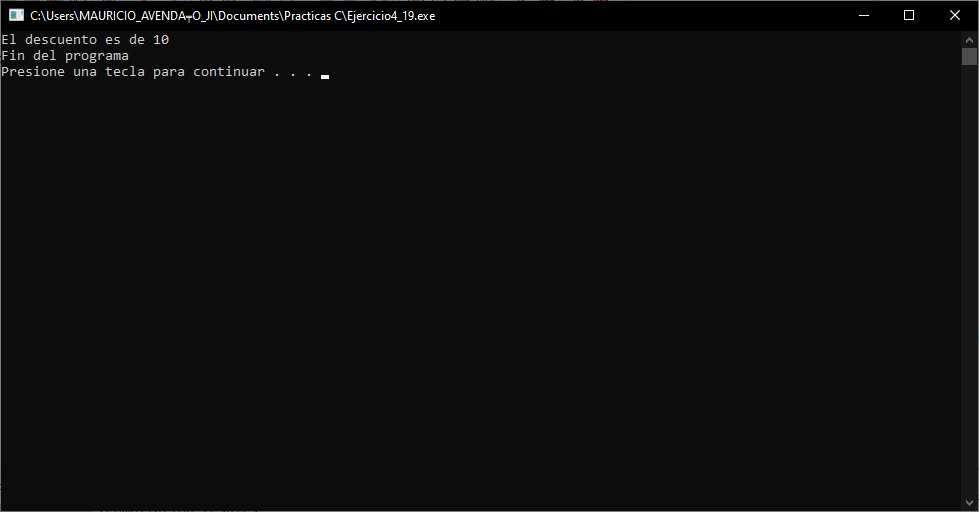
Declaración de variables, de tipo entero llamadas descuento, descuentoCliente, cliente y z, cada una con un valor, cabe mencionar que el valor que adoptan cliente y z, aunque son de tipo int toman un valor booleano, en este caso 0 corresponde a false, si fuera 1 seria true.

Método principal

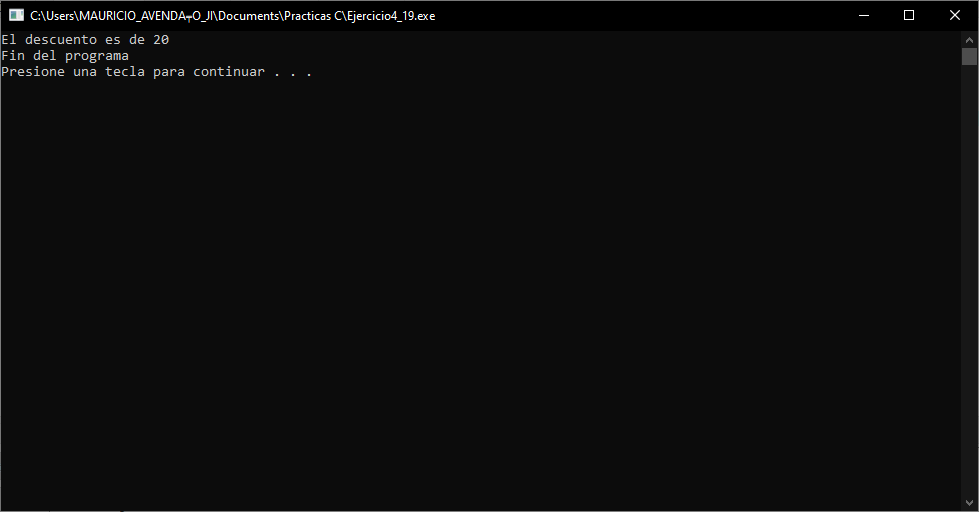
Función que nos permite detener la ejecución del programa y visualizar los resultados en pantalla.

Valor que devuelve el método principal

Como nos podemos dar cuenta, el valor de cliente es 0, es decir, falso y como la indicación del operador ternario dice que si es falso toma el tercer enunciado, entonces z adopta el valor de 10, que es descuento.



En el caso que el valor de cliente fuera 1, es decir, true el valor que z adoptaría seria el de descuentoCliente que es 20, porque la indicación del operador ternario dice que si el enunciado 1 es verdadero toma el valor del 2.





Ciclo while, la función de este es ejecutar cierto número de instrucciones mientras la condición se siga cumpliendo, es decir, tenemos a contador que vale 0, entonces entra al ciclo y pregunta, 0 es menor que 3, como es cierto imprime Hola, e incrementa 1, ahora vale 1, entra de nuevo a la condición y pregunta 1 es menor que 3, es cierto, entonces imprime otra vez Hola, incrementa de nuevo, ahora vale 2, entra otra vez a la condición y pregunta, 2 es menor que 3, es cierto, imprime otra vez Hola, incrementa 1, ahora vale 3, vuelve a la condición y pregunta 3 es menor que 3, falso, como no es cierto sale del ciclo. Imprime el mensaje que esta fuera del ciclo y termina el programa.

Como nos dimos cuenta se deben imprimir 3 Hola, verifiquemos.

Valor que devuelve el método principal

Función que nos permite detener la ejecución del programa y visualizar los resultados en pantalla.

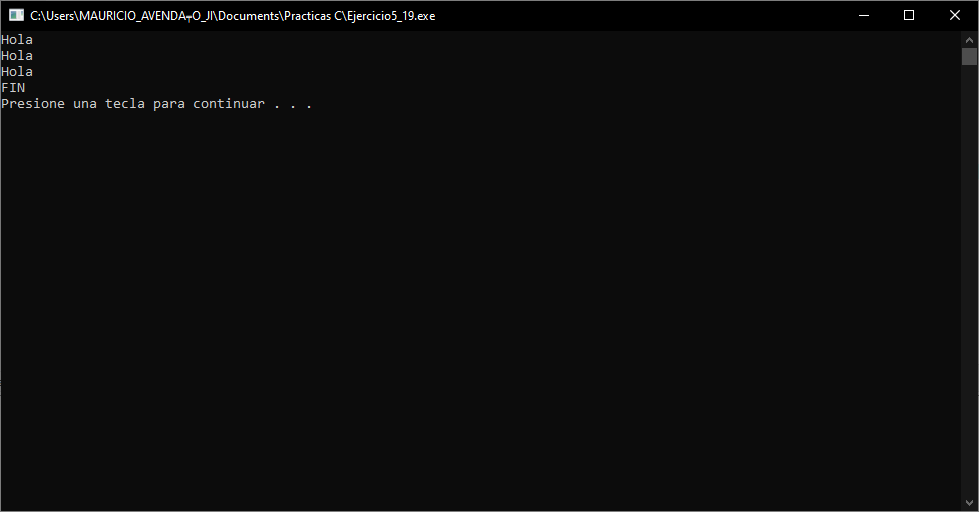
Incremento

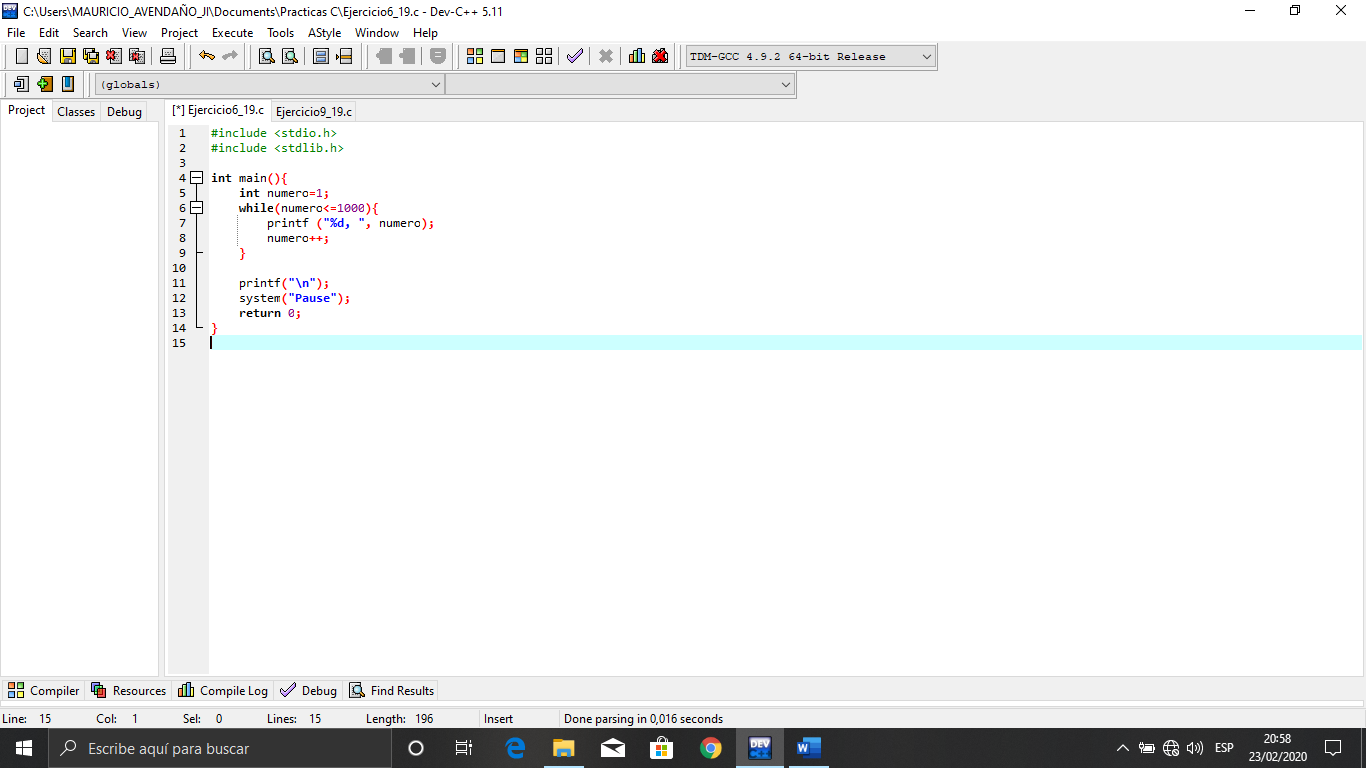
Condición

Declaración del contador de tipo entero, inicializado en 0.

Método principal

Bibliotecas





Valor que devuelve el método principal

Ciclo while, la función de este es ejecutar cierto número de instrucciones mientras la condición se siga cumpliendo, es decir, tenemos a número que vale 1, entonces entra al ciclo y pregunta, 1 es menor o igual a 1000, igual no pero menor si,, entonces imprime el valor del número (1), e incrementa 1, ahora vale 2, entra de nuevo a la condición y pregunta 2 es menor o igual que 1000, igual no pero menor si, entonces imprime el 2, incrementa de nuevo, ahora vale 3, entra otra vez a la condición y pregunta, 3 es menor o igual a 1000, igual no pero menor si, entonces imprime a 3, incrementa 1,y así sucesivamente, hasta que el valor del número no sea menor ni igual a 1000, sale del ciclo y termina el programa.

Incremento

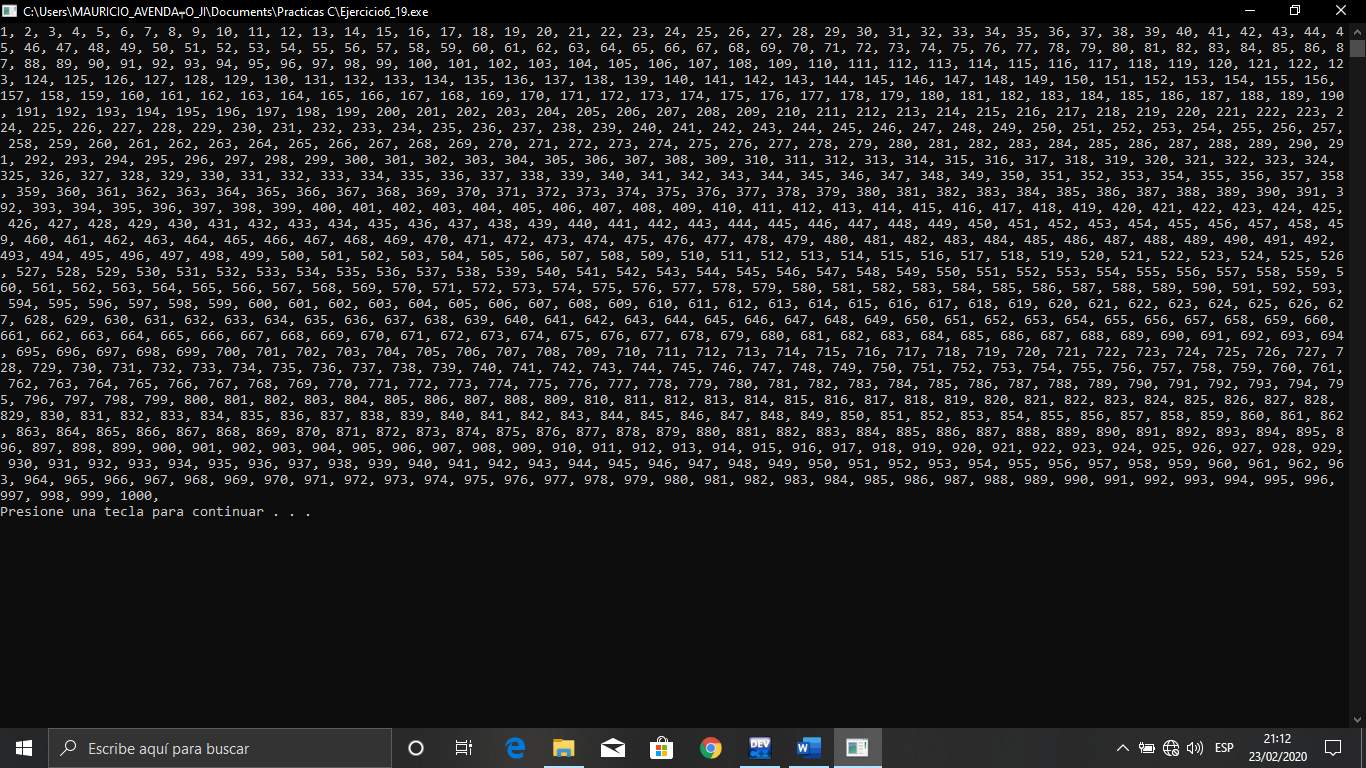
Condición

Declaración de la variable número de tipo entero, inicializado en 1.

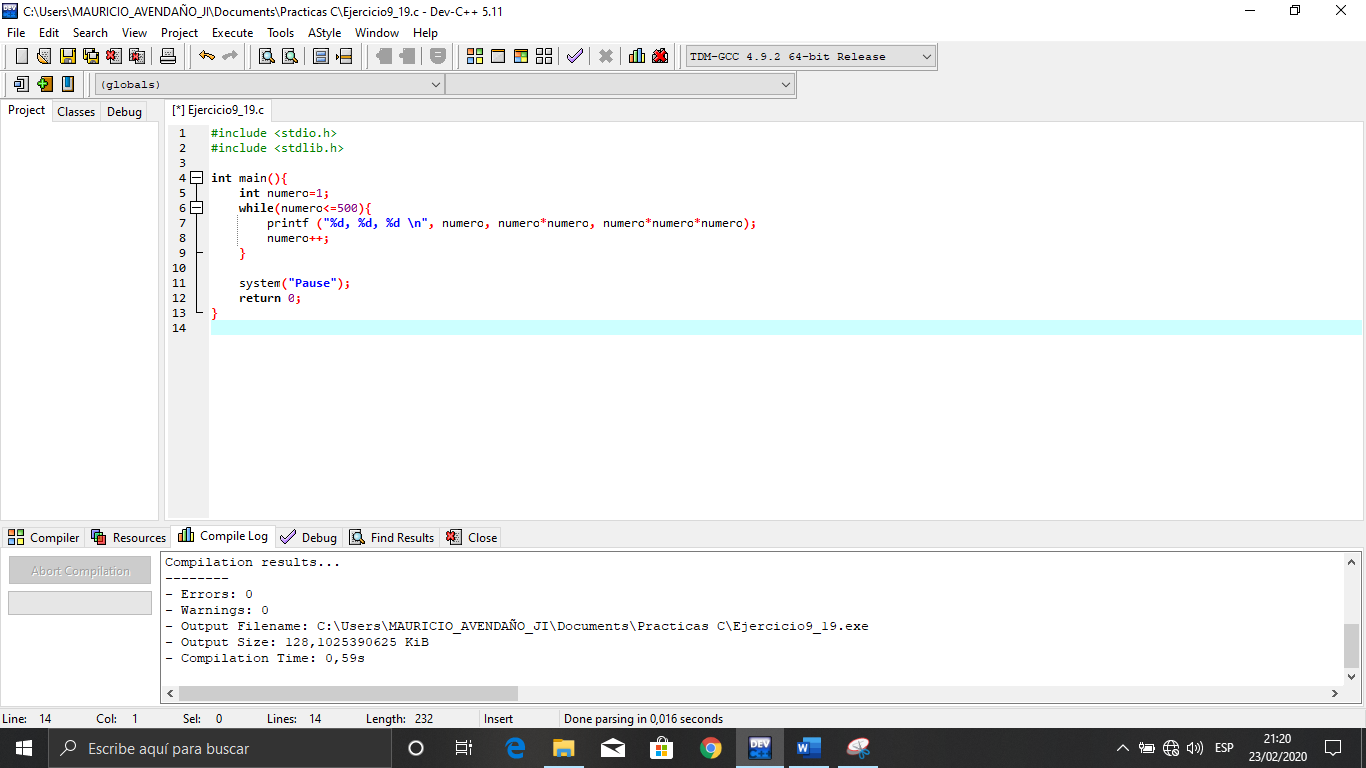
Método principal

Función que nos permite detener la ejecución del programa y visualizar los resultados en pantalla.

Bibliotecas

continua **…**





Método principal

v

Bibliotecas

Ciclo while, la función de este es ejecutar cierto número de instrucciones mientras la condición se siga cumpliendo, es decir, tenemos a número que vale 1, entonces entra al ciclo y pregunta, 1 es menor o igual a 500, igual no pero menor si, entonces imprime el número, el cuadrado y el cubo del mismo, resultado de las operaciones, e incrementa 1, ahora vale 2, entra de nuevo a la condición y pregunta 2 es menor o igual que 500, igual no pero menor si, entonces imprime otra vez, el número, su cuadrado y su cubo, incrementa de nuevo, ahora vale 3, entra otra vez a la condición y pregunta, 3 es menor o igual a 500, igual no pero menor si, entonces imprime al número, su cuadrado y su cubo, incrementa 1,y así sucesivamente, hasta que el valor del número no sea menor ni igual a 500, sale del ciclo y termina el programa.

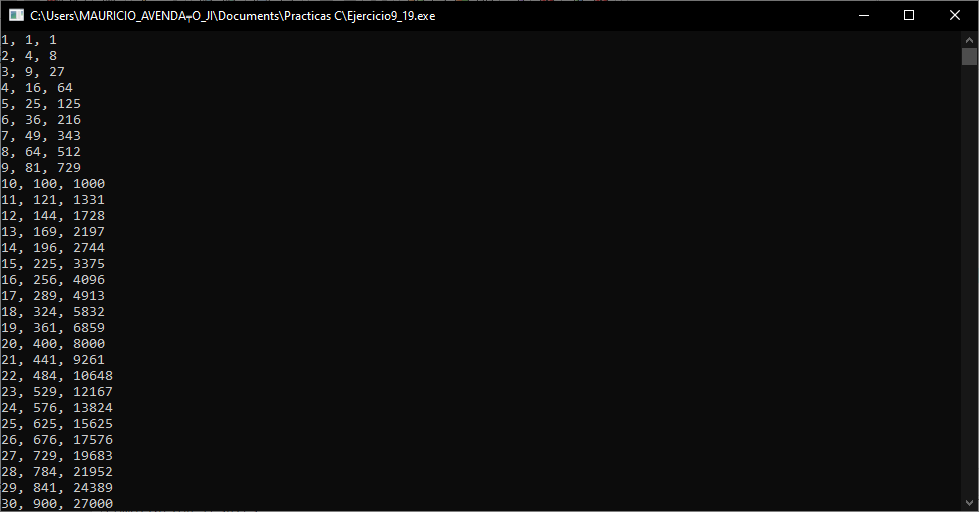
Condición

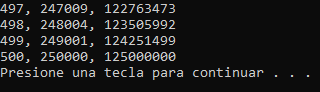
Incremento

Valor que devuelve el método principal

Declaración del contador de tipo entero, inicializado en 0.

Función que nos permite detener la ejecución del programa y visualizar los resultados en pantalla.

 continua**…**

**…**

**V. Conclusiones:**

Se puede finalizar diciendo que C es un lenguaje que nos permite realizar codificaciones en algunas veces mejores que en algunos otros lenguajes, pero siempre y cuando se aprenda como realizar estos, para hacerlo tenemos que iniciar con las sintaxis básicas de programación de este lenguaje, por ejemplo, la declaración y uso de variables y constantes, las sentencias condicionales, las sentencias repetitivas.

Todo esto se aprende llevándolo a práctica, pero de igual manera todo lleva su tiempo.

con estos ejercicios se puede aprender cosas importantes como, por ejemplo, que es una secuencia de escape, que es un especificador de conversión, los operadores de dirección, como ingresar datos por teclado, etc.