|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre de la práctica** | **EJERCICIOS UNIDAD II (Parte 3)** | | | **No.** | **3** |
| **Asignatura:** | **METODOS NUMERICOS** | **Carrera:** | **ING. SISTEMAS COMPUTACIONALES** | **Duración de la práctica (Hrs)** | **2** |

**BECERRA HERNÁNDEZ JESSICA LILIAN 342 ISIC**

**I. Competencia(s) específica(s):**

Utiliza algoritmos numéricos que proporcionen el mínimo de error para obtener soluciones

aproximadas de modelos matemáticos de aplicación en ingeniería que no puedan resolverse por métodos analíticos utilizando un lenguaje de programación como herramienta.

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

Otro

**III. Material empleado:**

DEV C++

Computadora

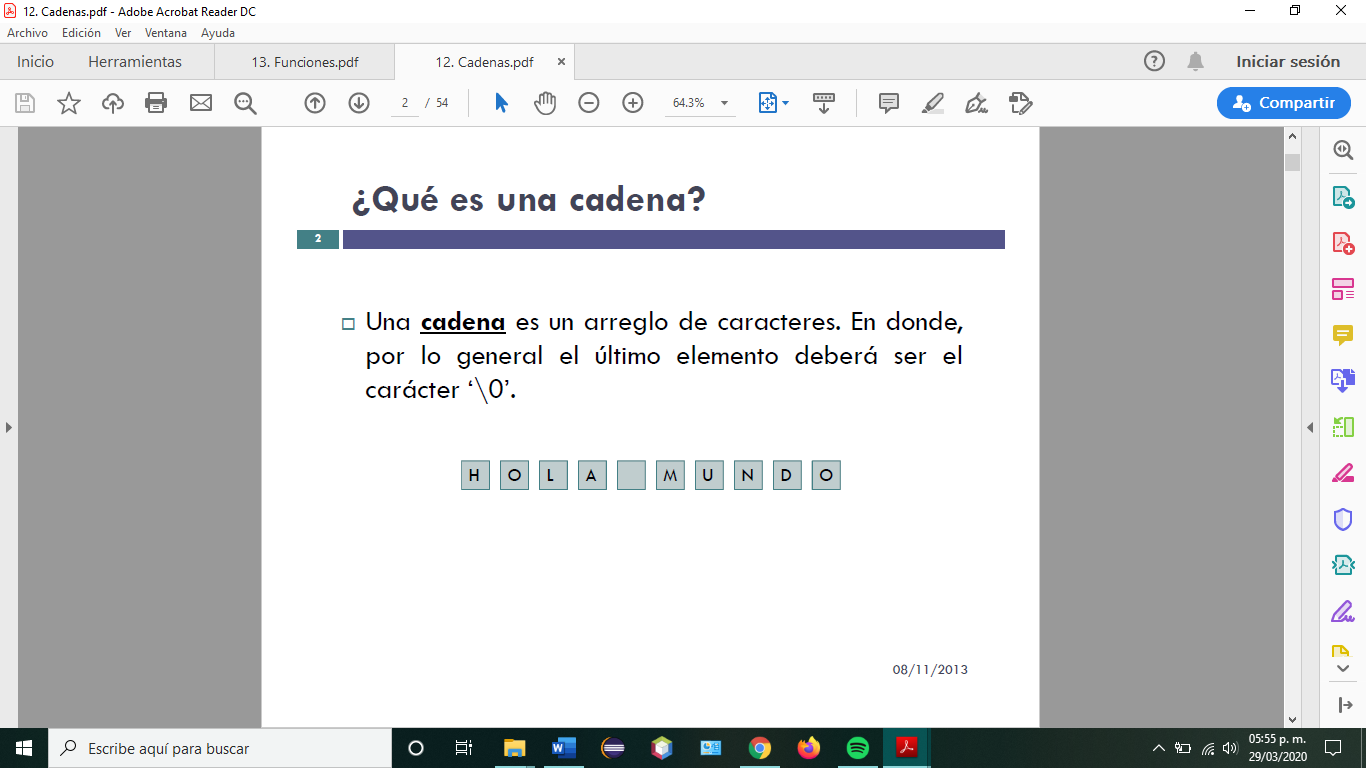
**IV. Desarrollo de la práctica:**

EJERCICIO 57

Cadenas

Una cadena es un arreglo de caracteres. En donde, por lo general el último elemento deberá ser el carácter ‘\0’.

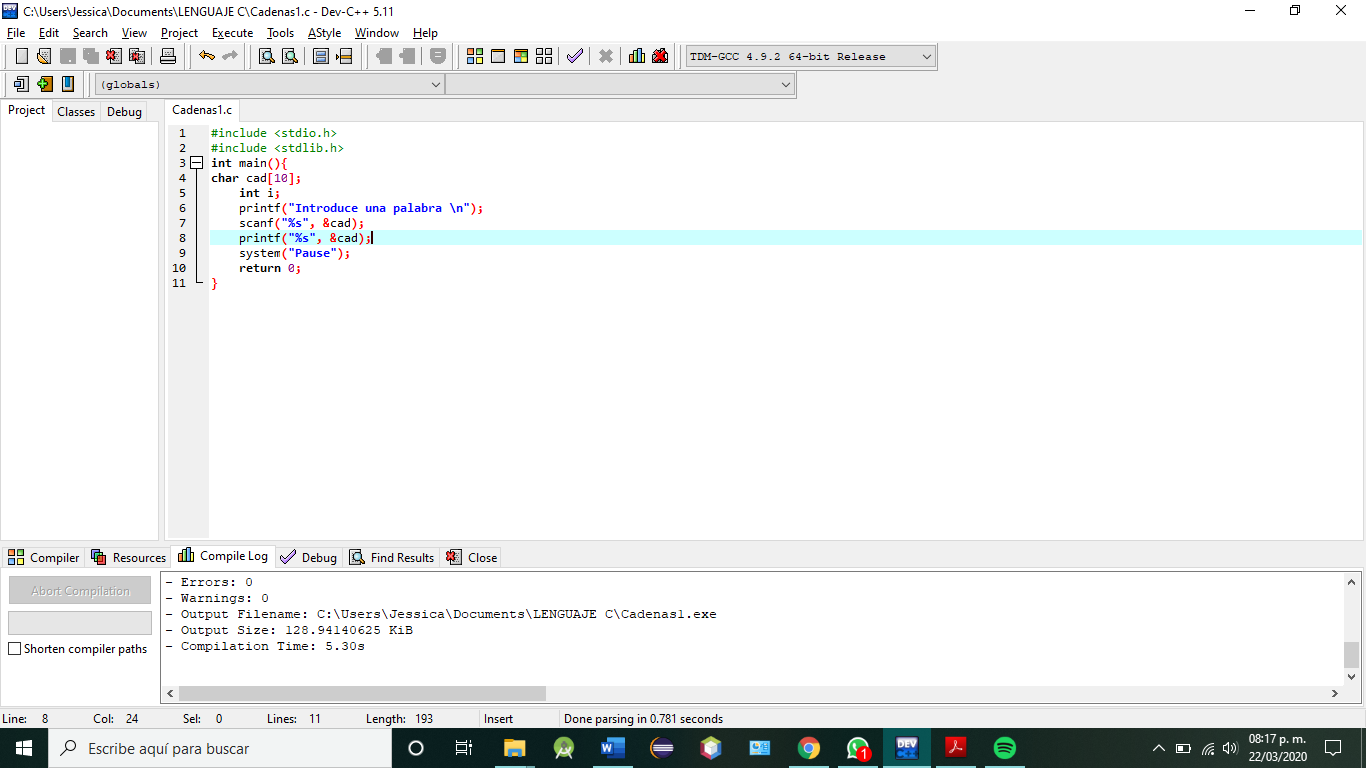
Ejemplo:



¿Cómo declarar un arreglo?

Crearla como un arreglo

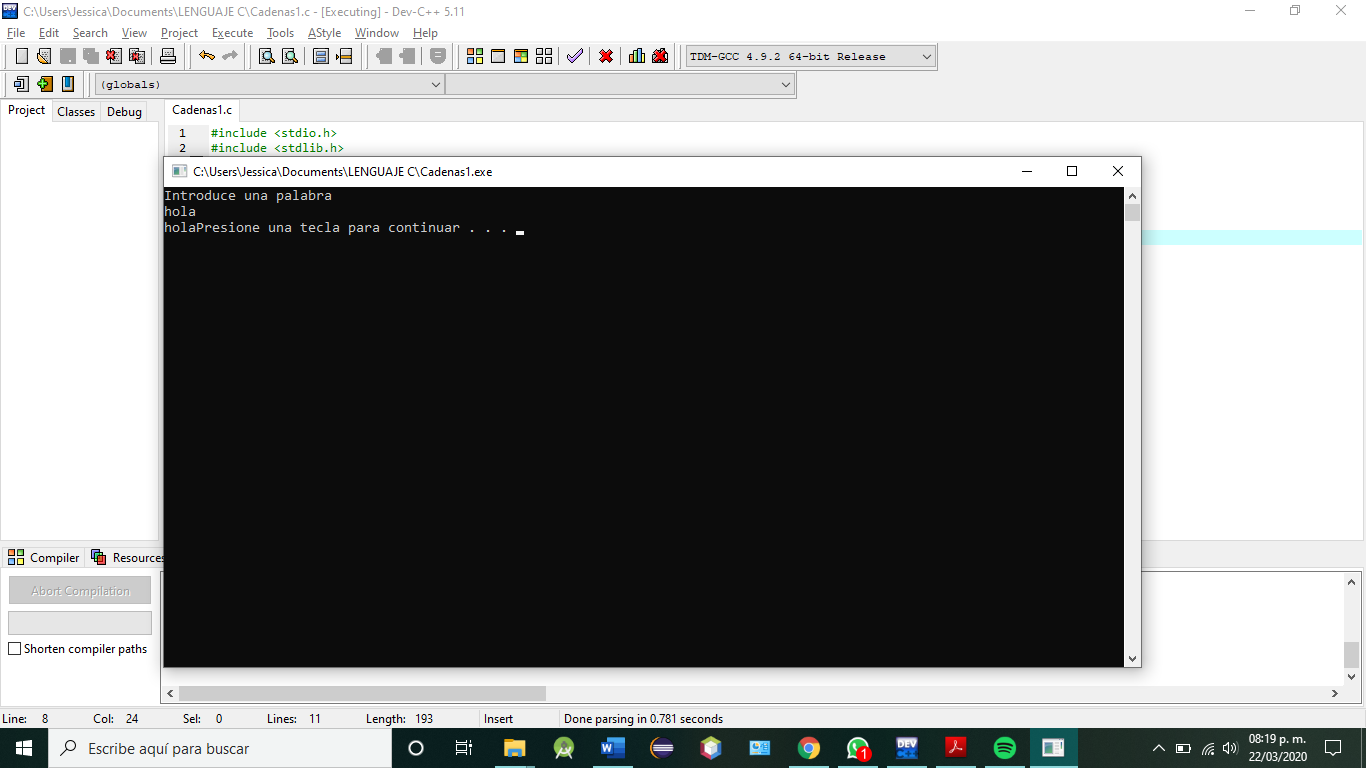
char c [10];



Damos clic en el botón de compilar y ejecutar, o usando la función F11.



Se nos muestra nuestro programa por consola.

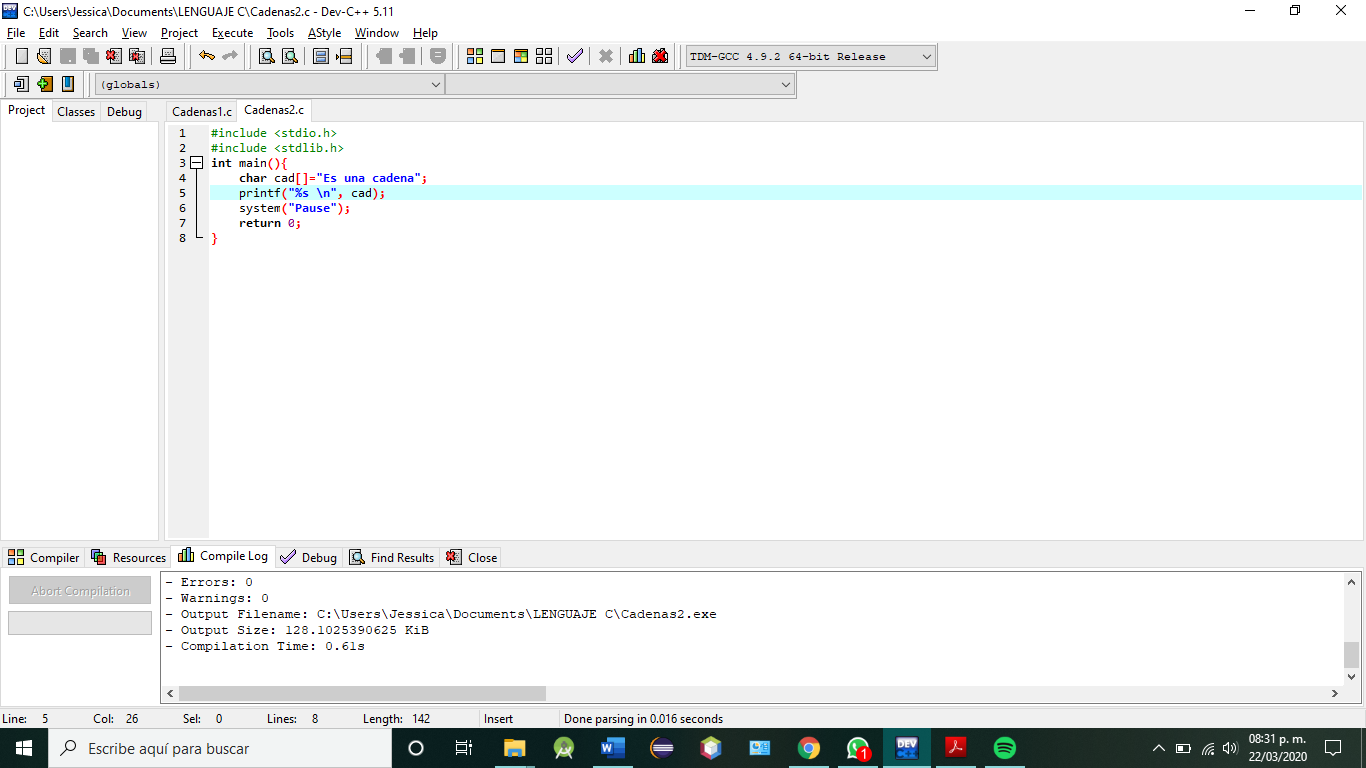


EJERCICIO 58

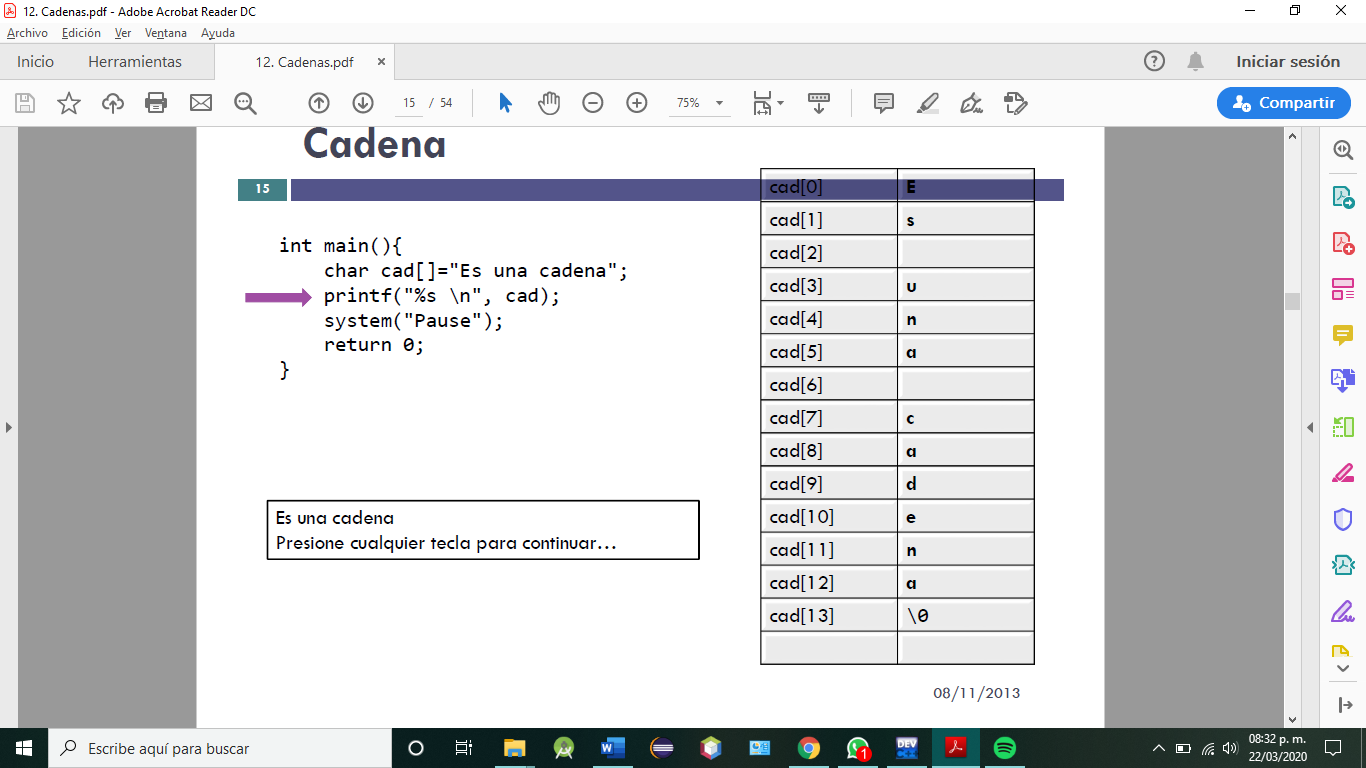
Crearla como arreglo sin tamaño

char cad [] = “Es una cadena”;

¿De cuántos elementos consta el arreglo cad? 14



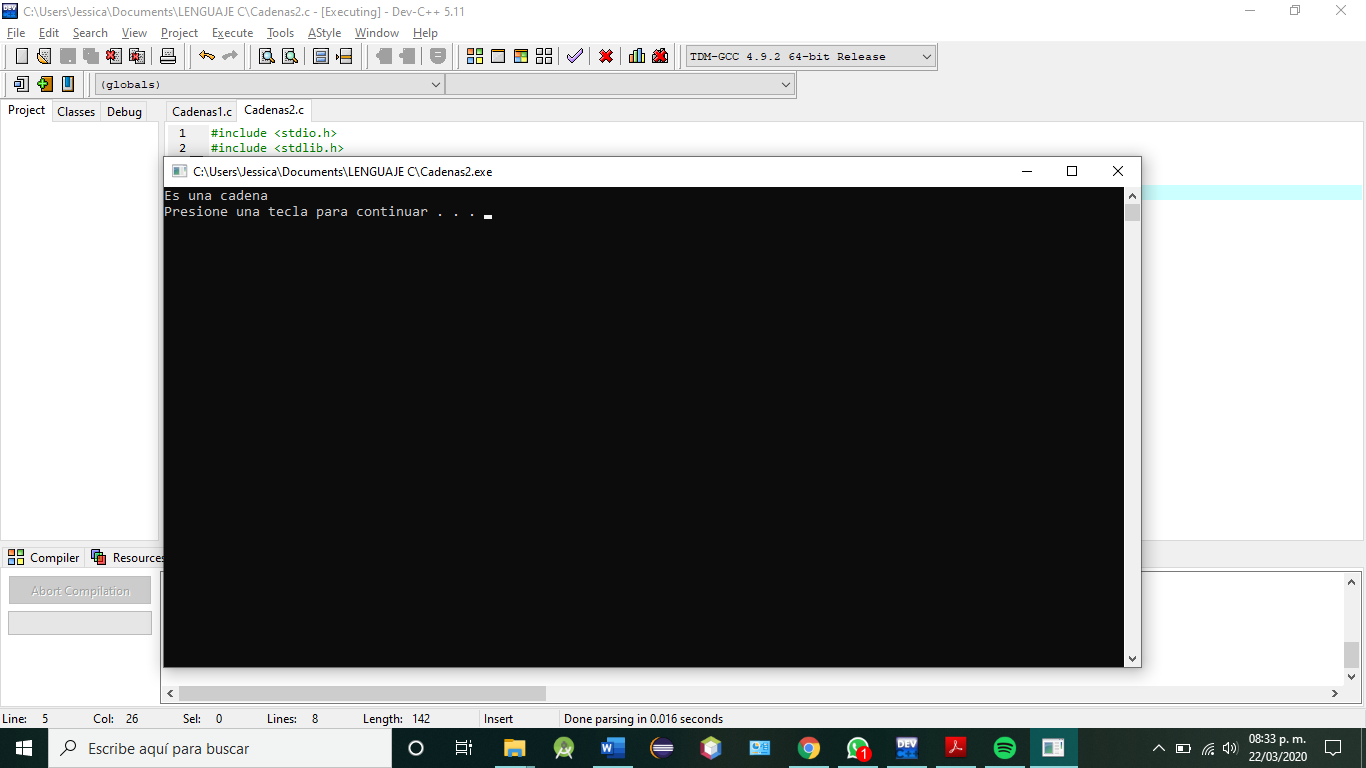
En memoria se comporta de la siguiente manera:



Damos clic en el botón de compilar y ejecutar, o usando la función F11.



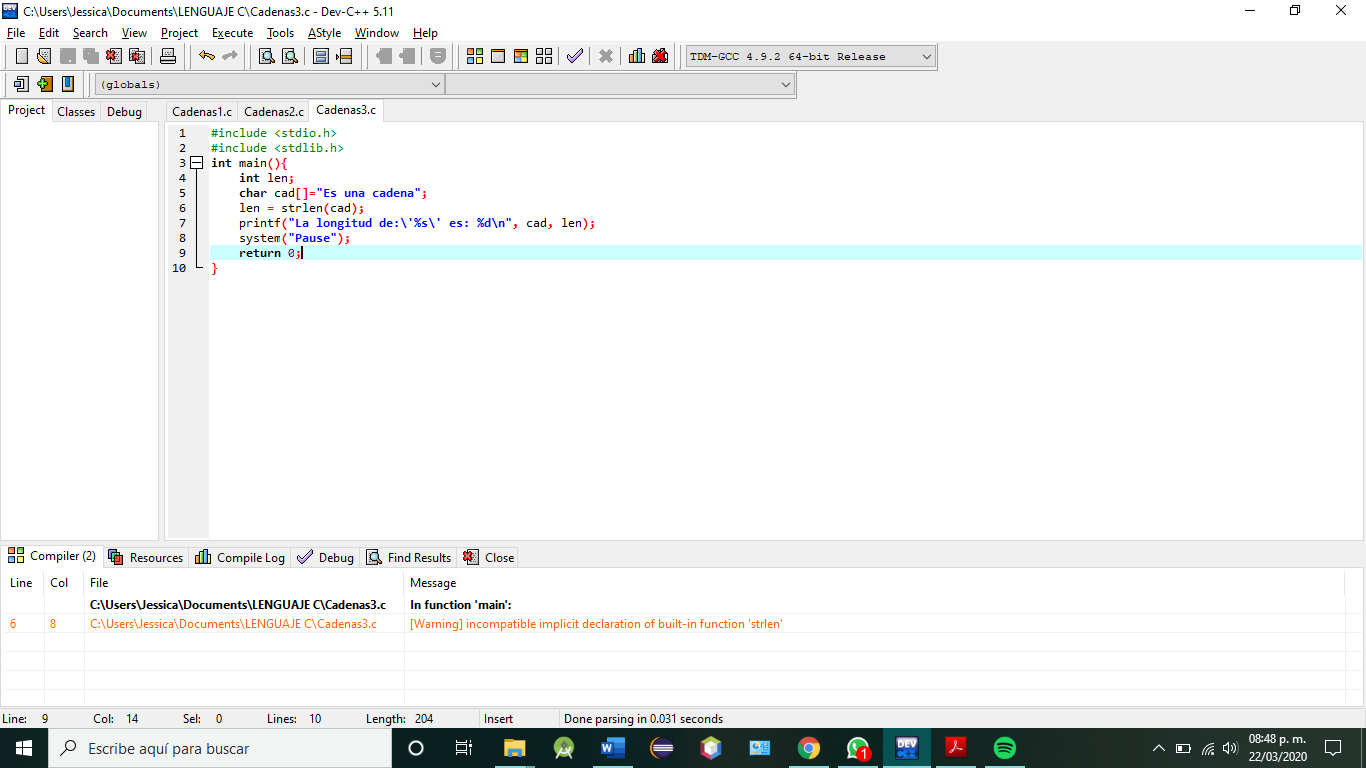
Se nos muestra nuestro programa por consola.



EJERCICIO 59

Función **Strlen**

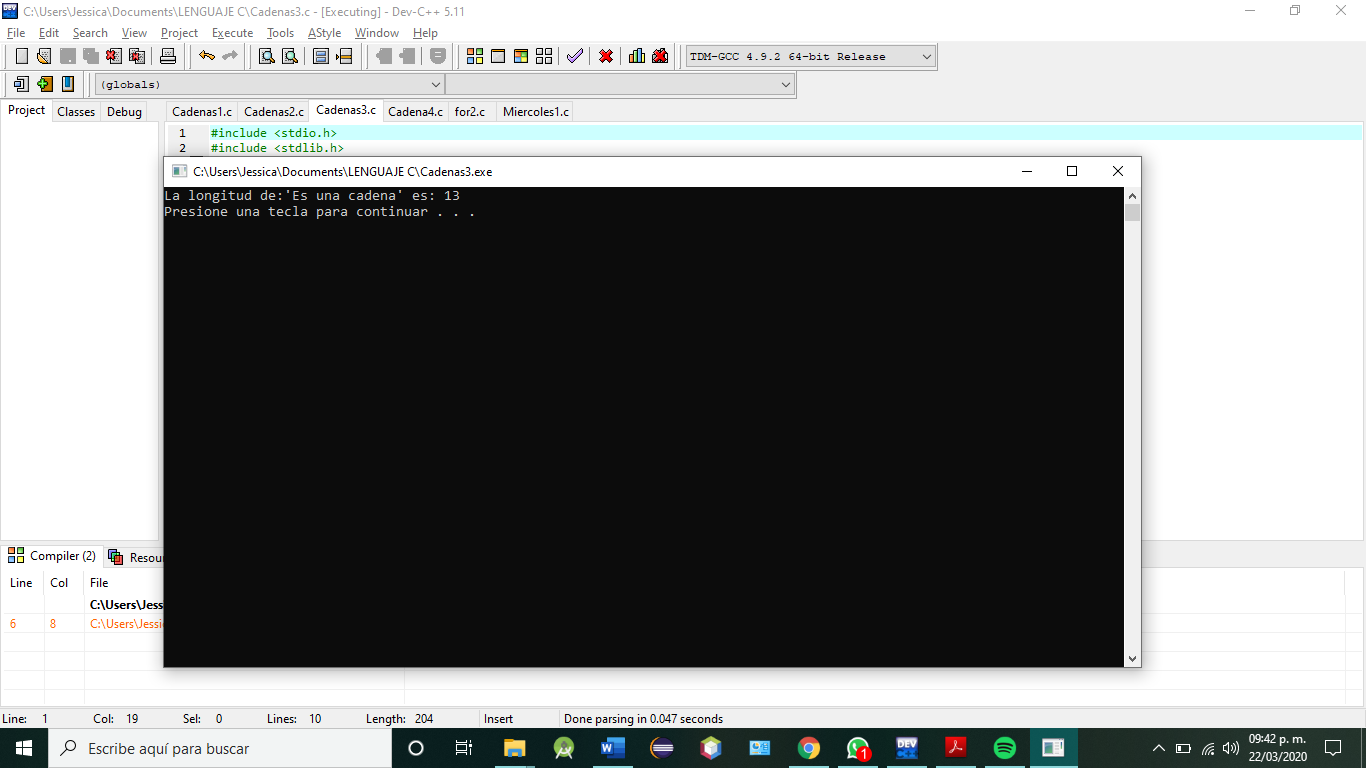
La función de **Strlen** devuelve la longitud de la cadena sin tomar en cuenta el carácter de final de cadena.



Damos clic en el botón de compilar y ejecutar, o usando la función F11.



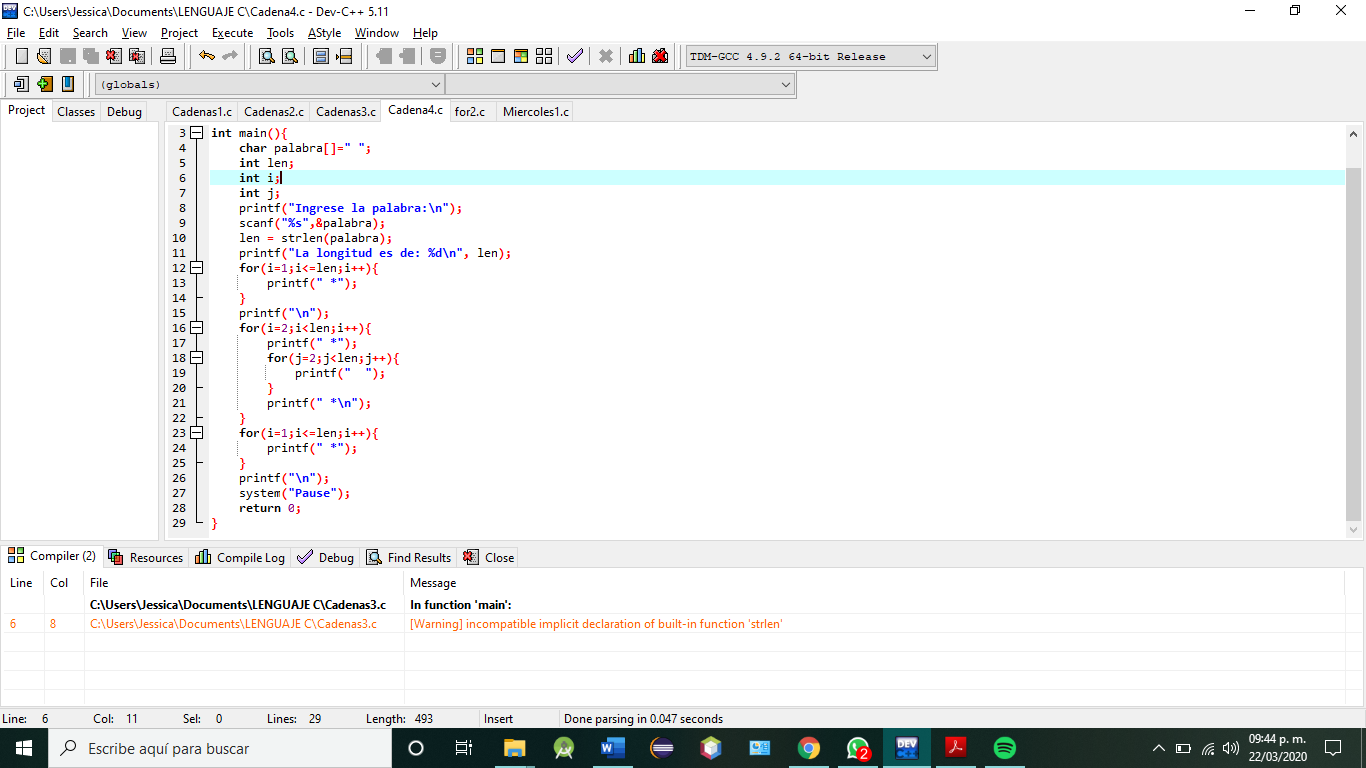
Se nos muestra nuestro programa por consola.



EJERCICIO 60

Escribe un programa que reciba una palabra por teclado.

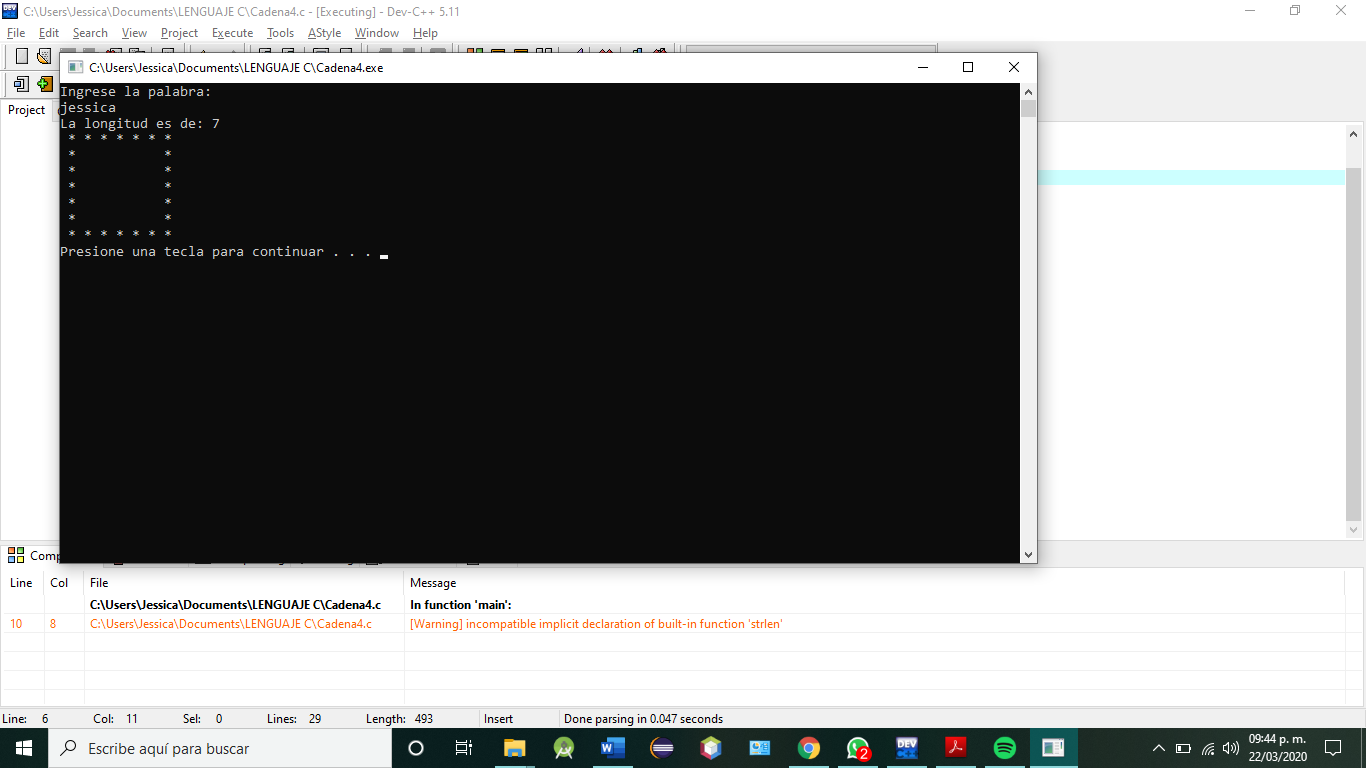
De acuerdo a la longitud de la palabra (N) que se ingresó por teclado imprime un cuadrado de asteriscos de (N x N).



Damos clic en el botón de compilar y ejecutar, o usando la función F11.



Se nos muestra nuestro programa por consola.

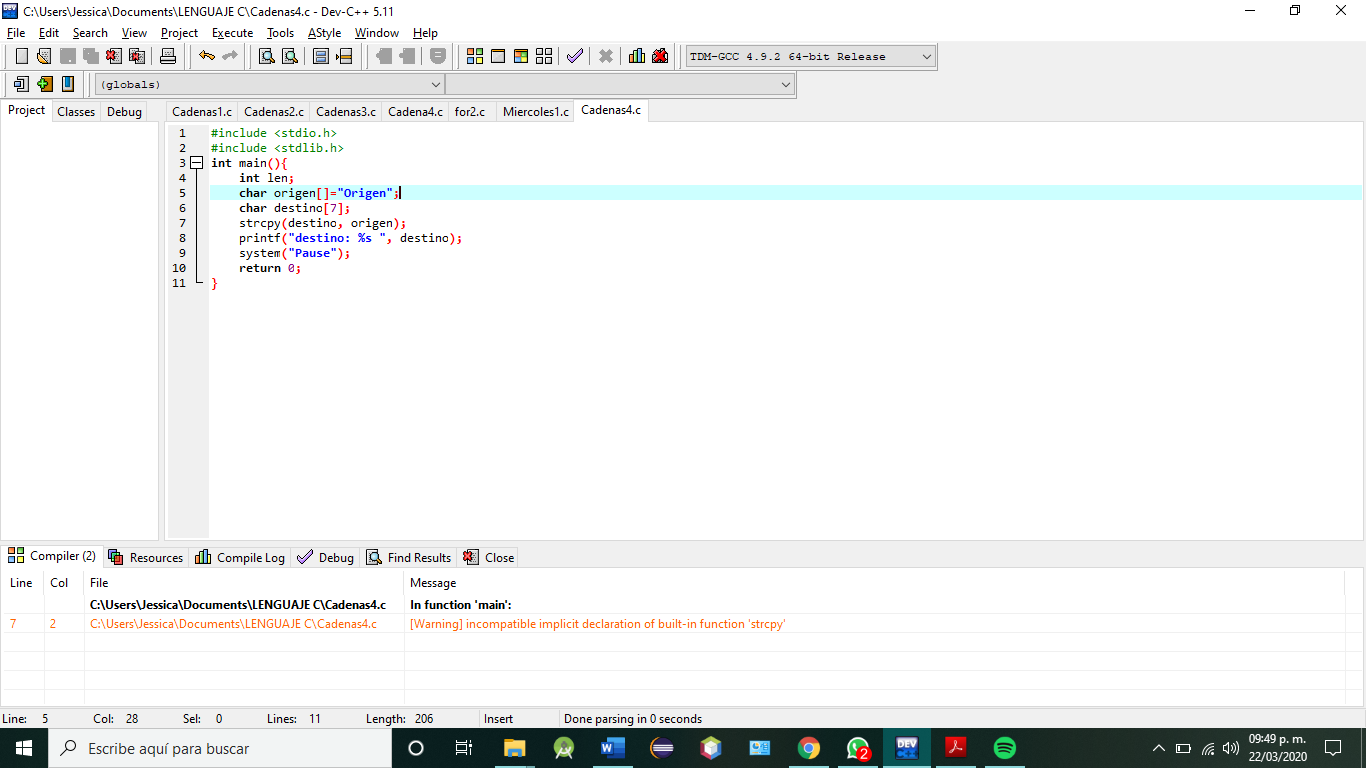


EJERCICIO 61

Copia el contenido de:

<cadena\_origen> en <cadena\_destino>.

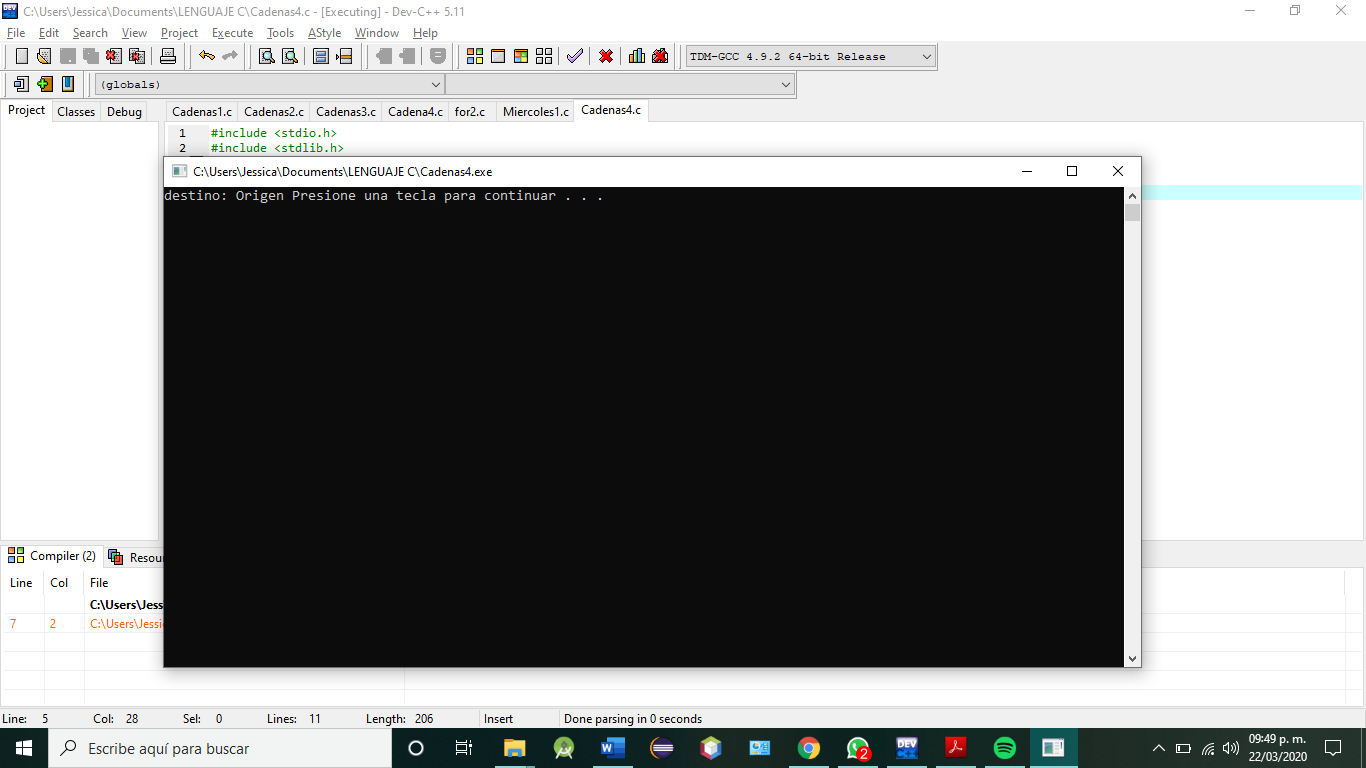
strcpy(<cadena\_destino>, <cadena\_origen>)



Damos clic en el botón de compilar y ejecutar, o usando la función F11.



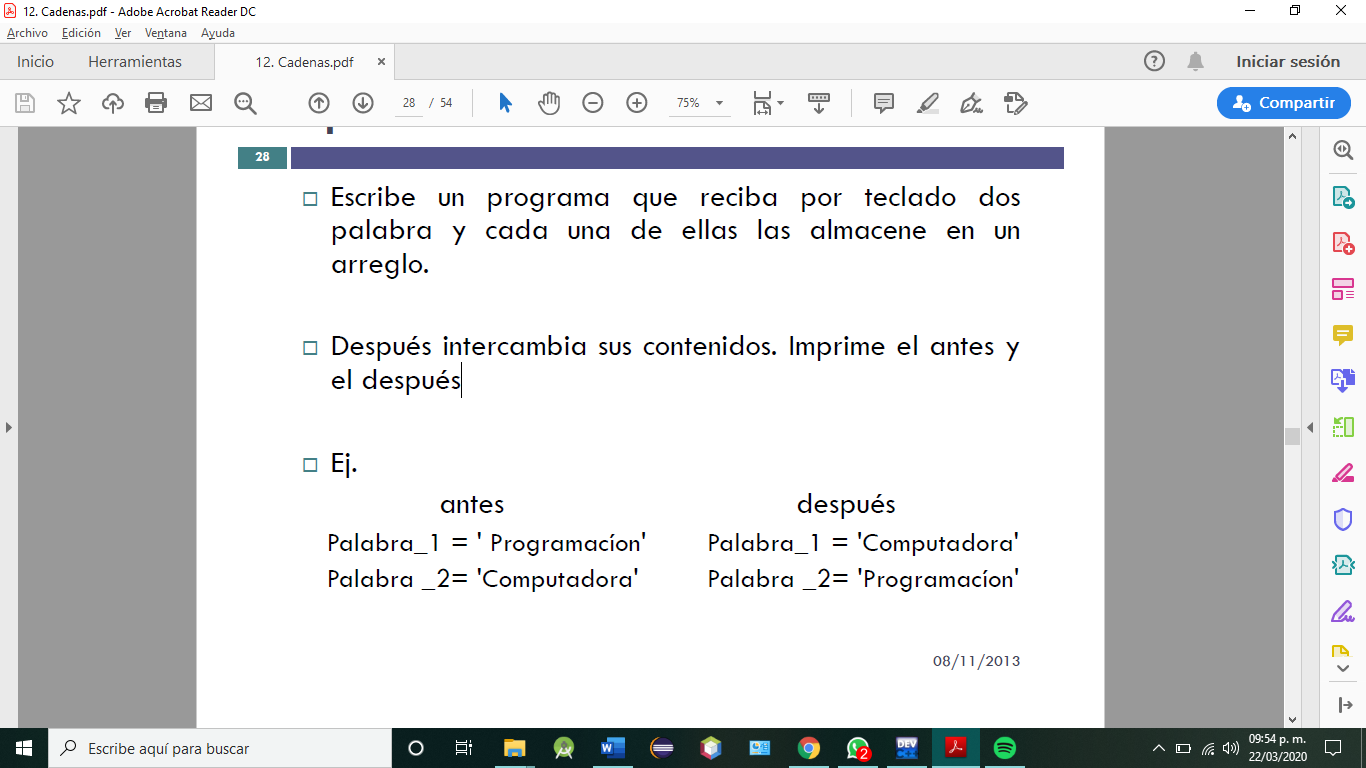
Se nos muestra nuestro programa por consola.

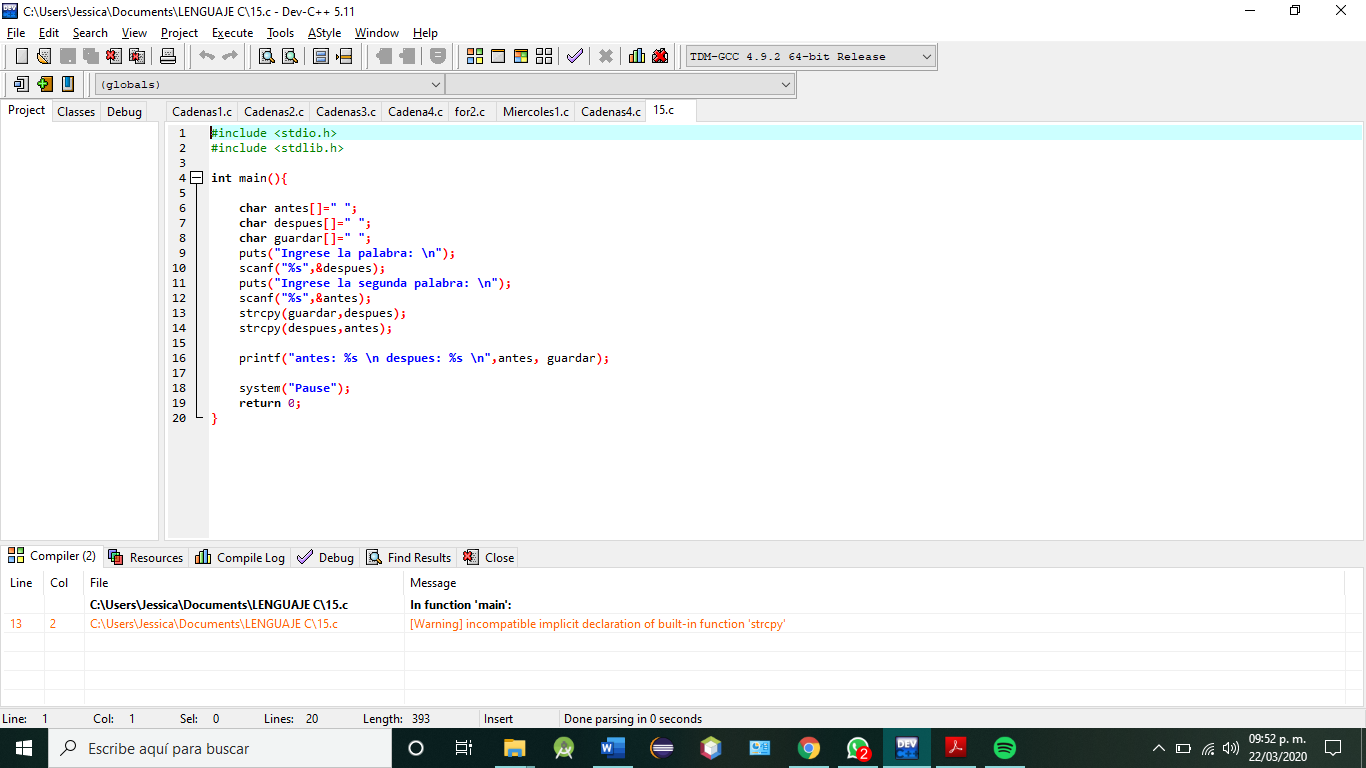


EJERCICIO 62

Escribe un programa que reciba por teclado dos palabras y cada una de ellas las almacene en un arreglo.

Después intercambia sus contenidos. Imprime el antes y el después.

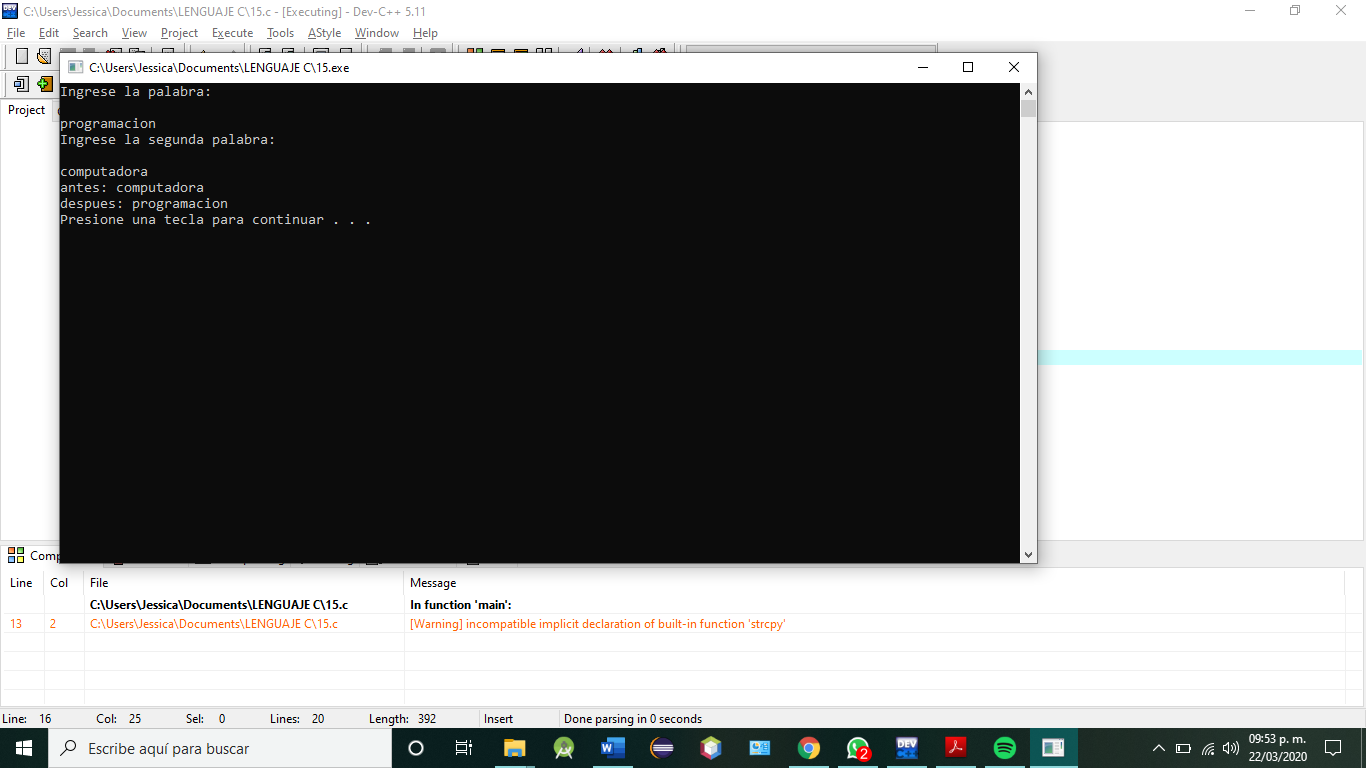




Damos clic en el botón de compilar y ejecutar, o usando la función F11.



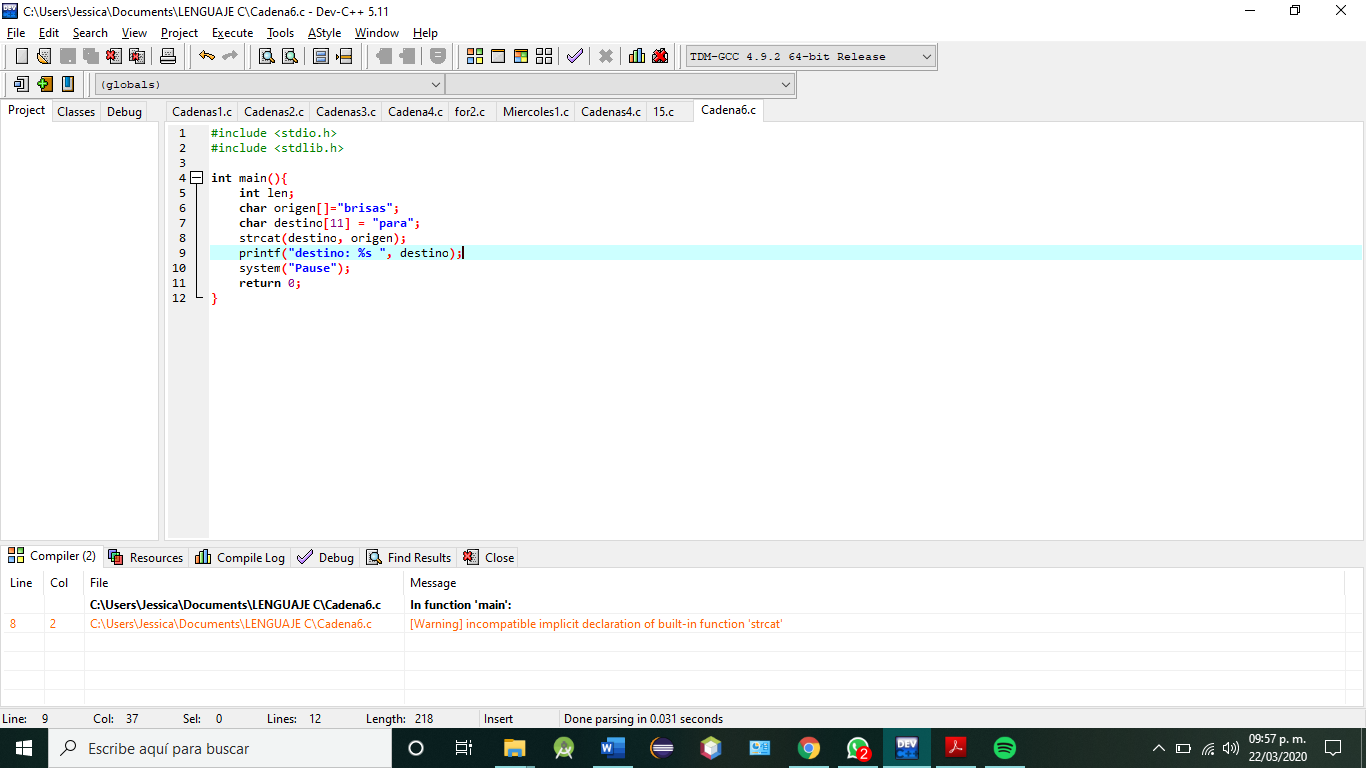
Se nos muestra nuestro programa por consola.



EJERCICIO 63

Concatena el contenido de <cadena\_origen> al final de <cadena\_destino>

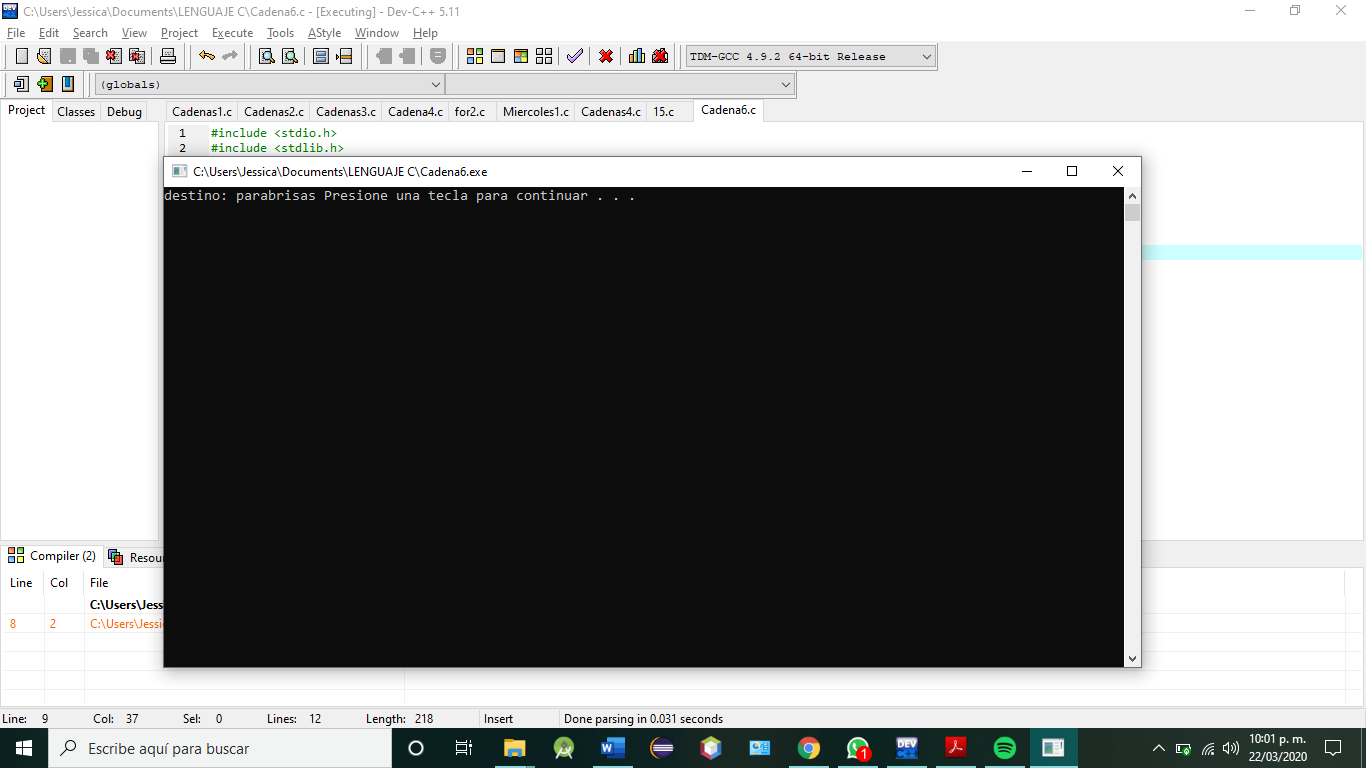
strcat(<cadena\_destino>, <cadena\_origen>)



Damos clic en el botón de compilar y ejecutar, o usando la función F11.



Se nos muestra nuestro programa por consola.

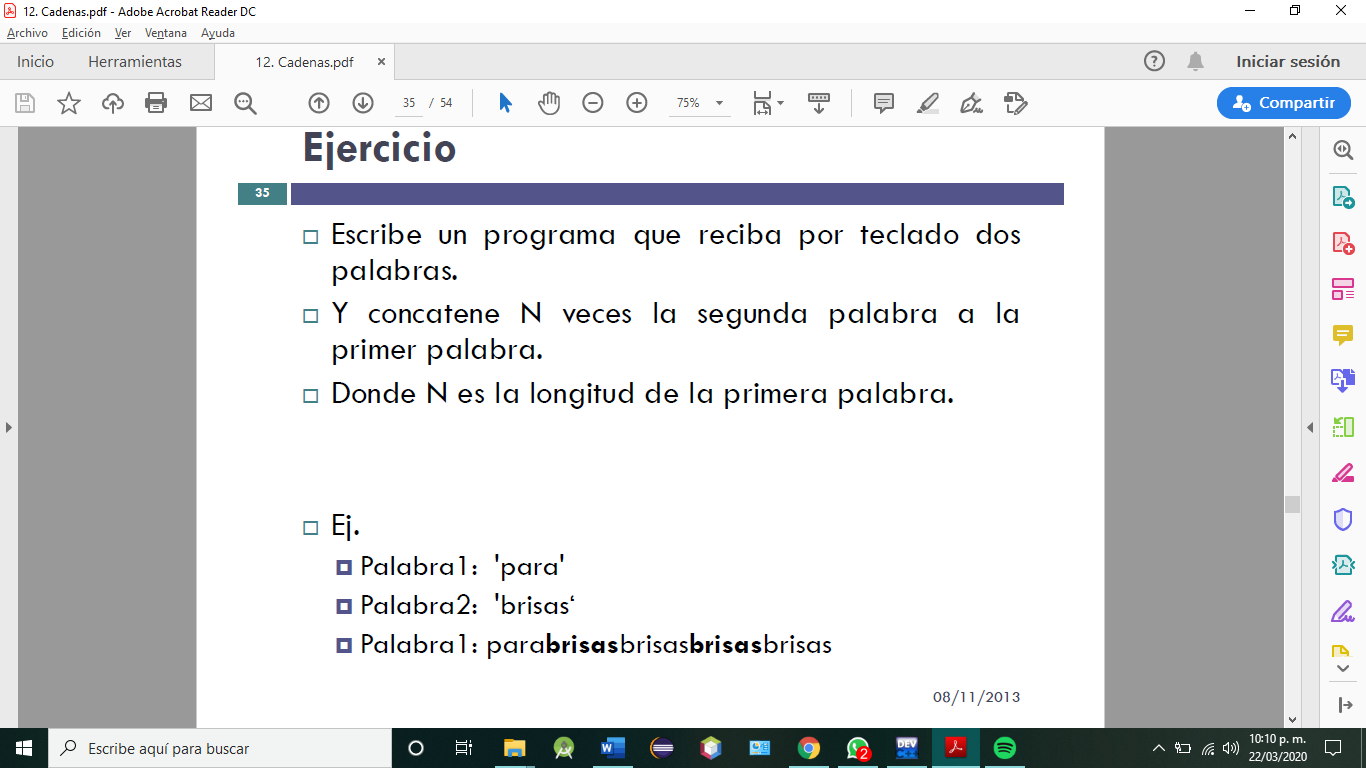


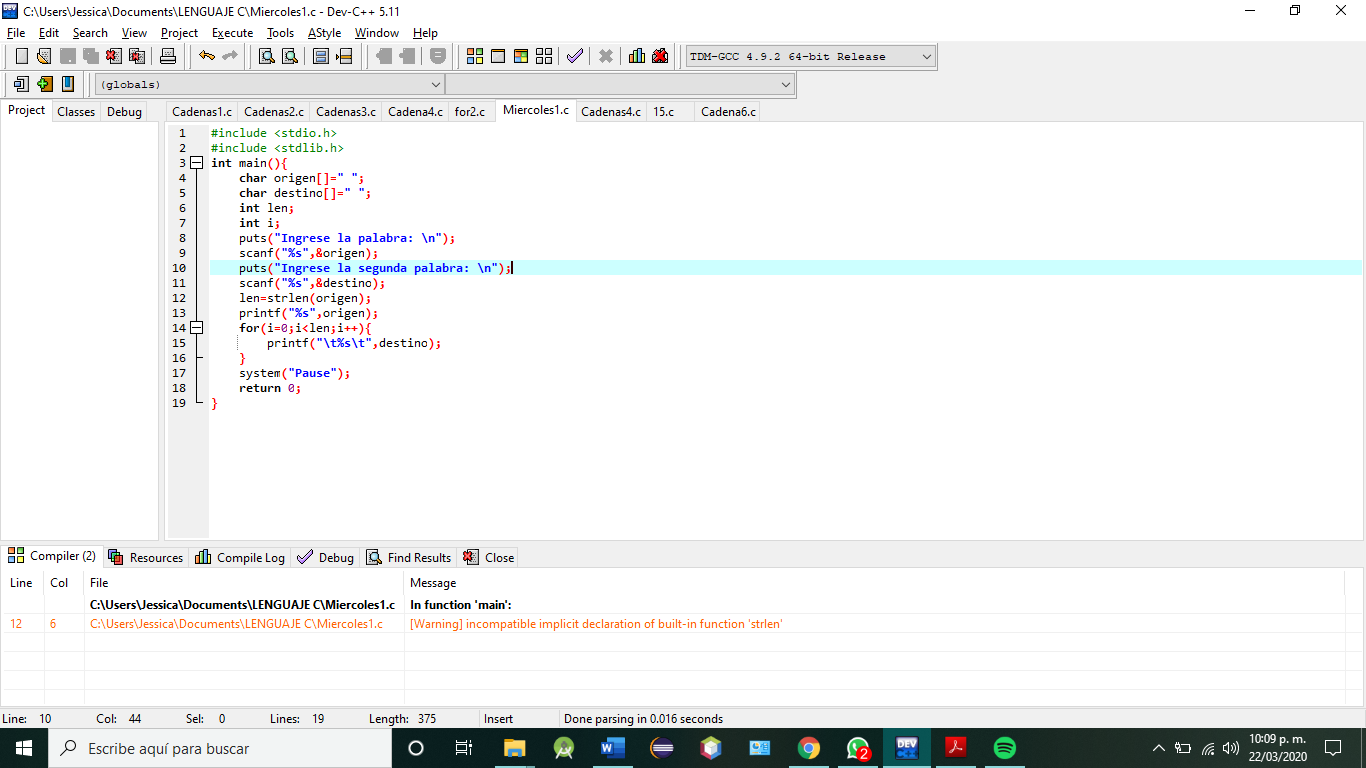
EJERCICIO 64

Escribe un programa que reciba por teclado dos palabras.

Y concatene N veces la segunda palabra a la primera palabra.

Donde N es la longitud de la primera palabra.

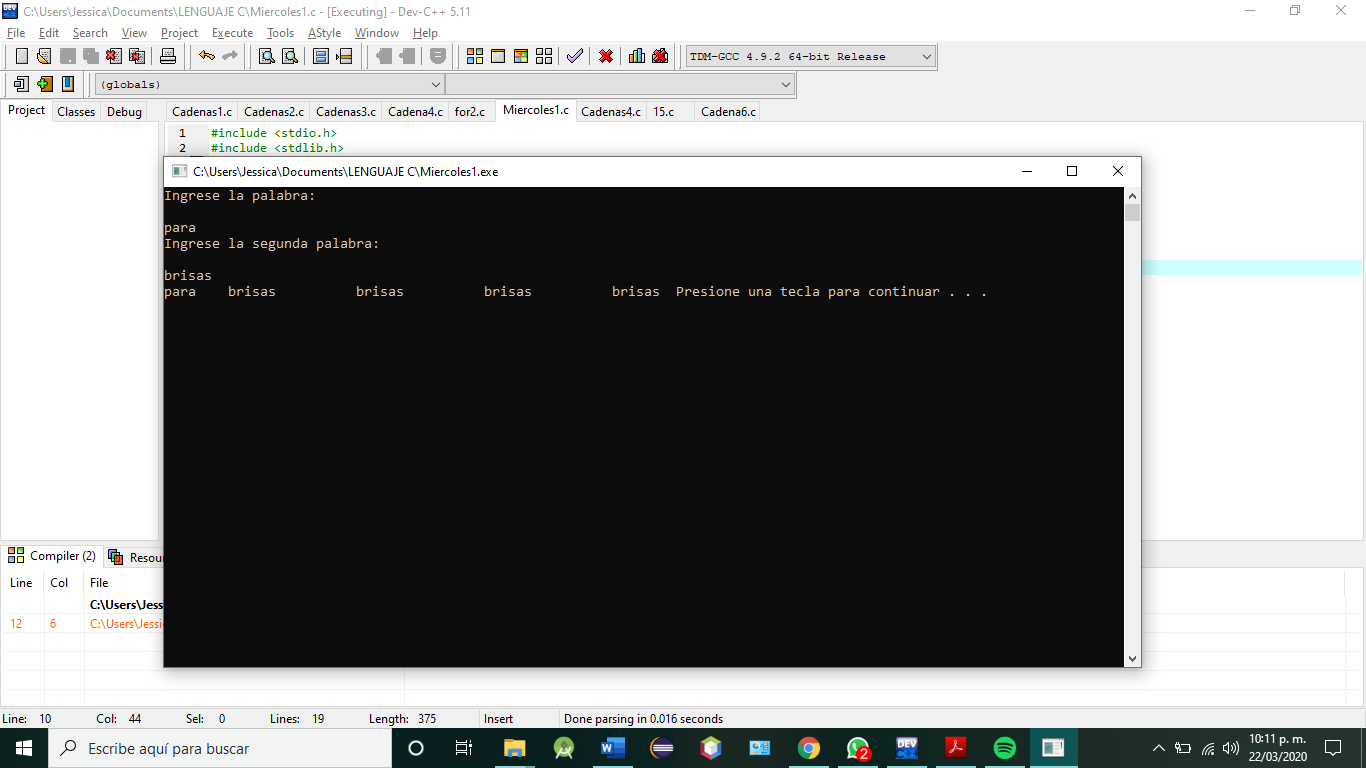




Damos clic en el botón de compilar y ejecutar, o usando la función F11.



Se nos muestra nuestro programa por consola.



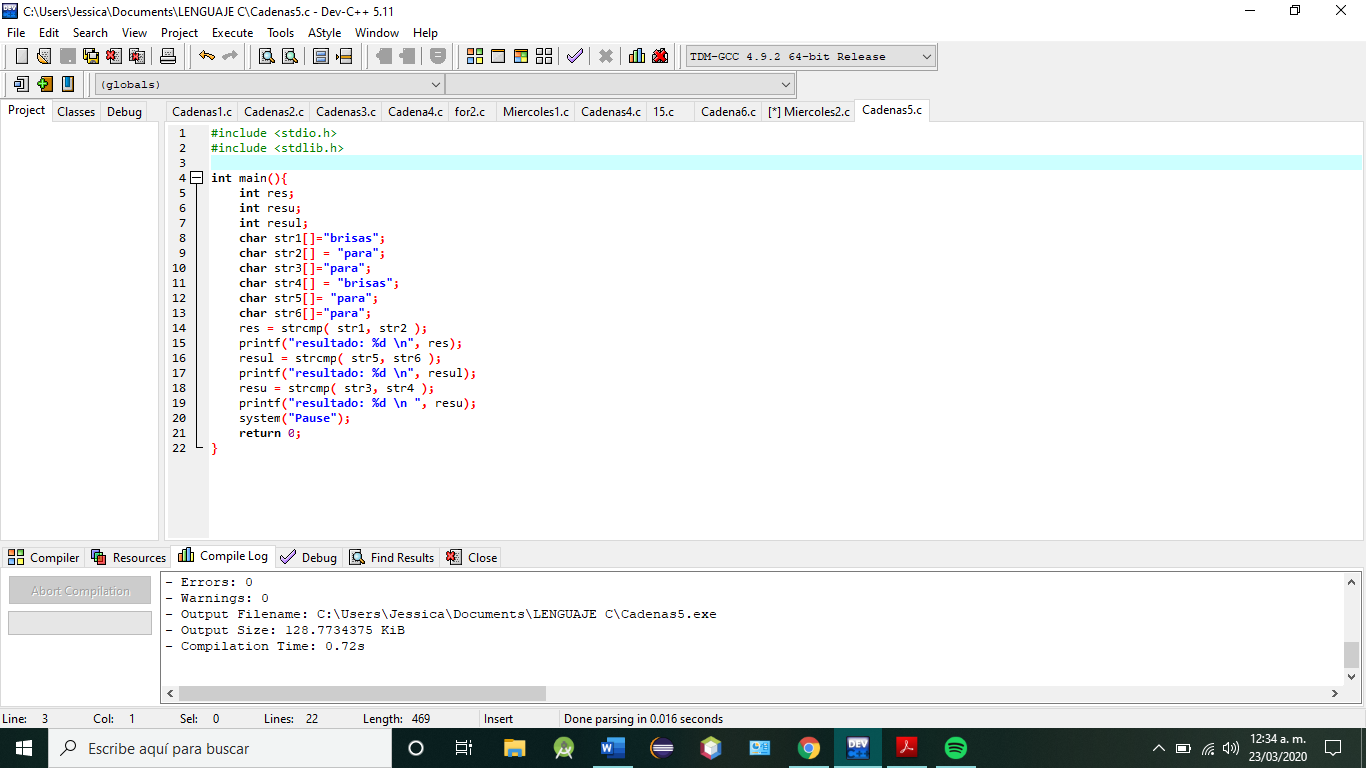
EJERCICIO 65

Compara las dos cadenas y devuelve un 0 si las dos cadenas son iguales.

Un número negativo si <cadena1> es menor que <cadena2>

Un número positivo (mayor que cero) si <cadena1> es mayor que <cadena2>.

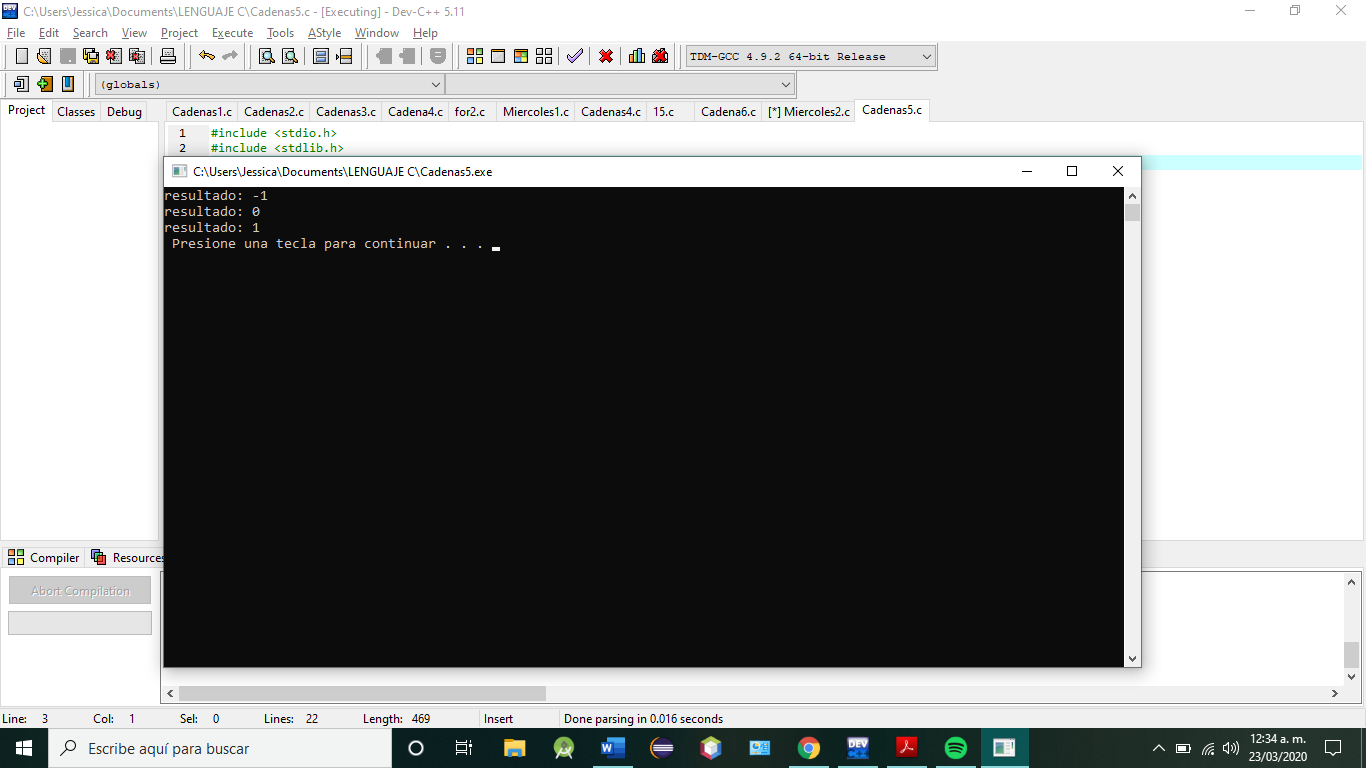
strcmp(<cadena1>, <cadena2>)



Damos clic en el botón de compilar y ejecutar, o usando la función F11.



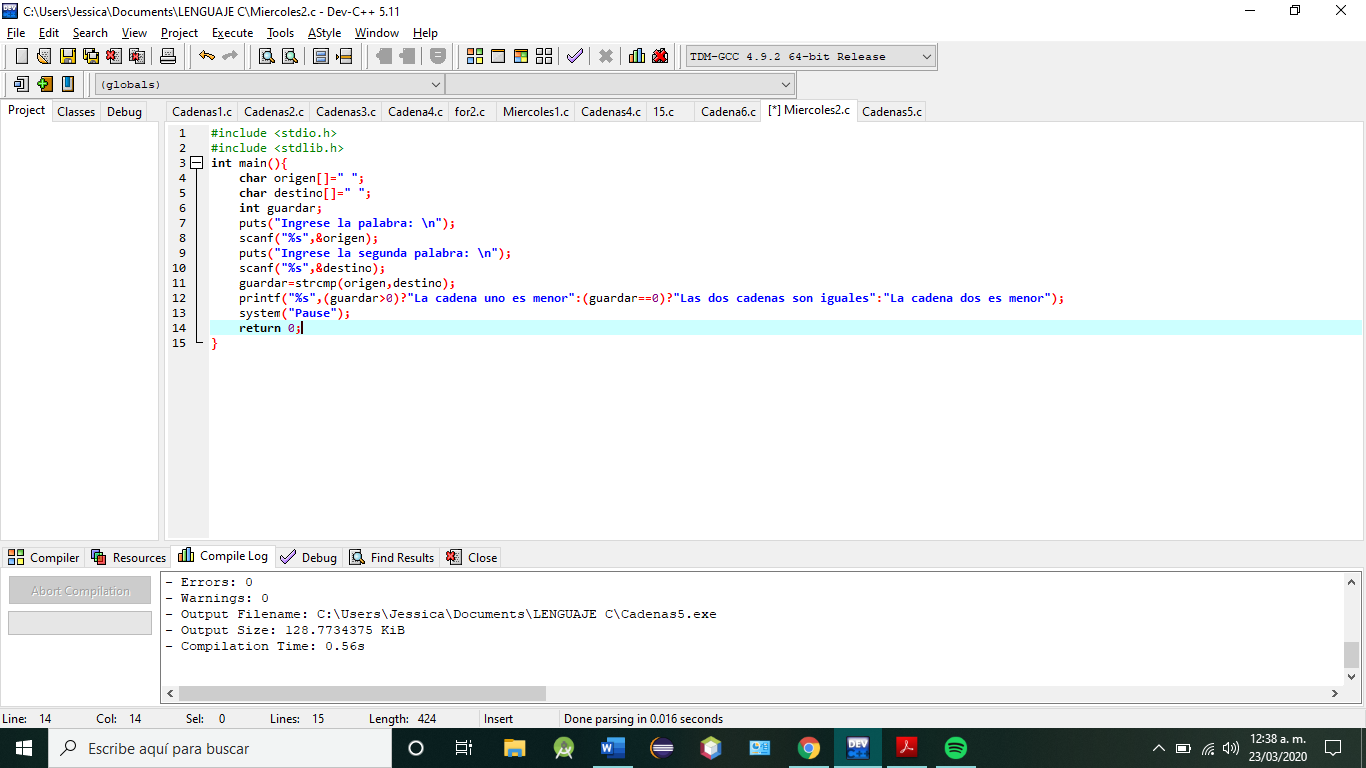
Se nos muestra nuestro programa por consola.



EJERCICIO 66

Escribe un programa que reciba por teclado dos palabras y te indique cuál de ellas es mayor y cuál es la menor.

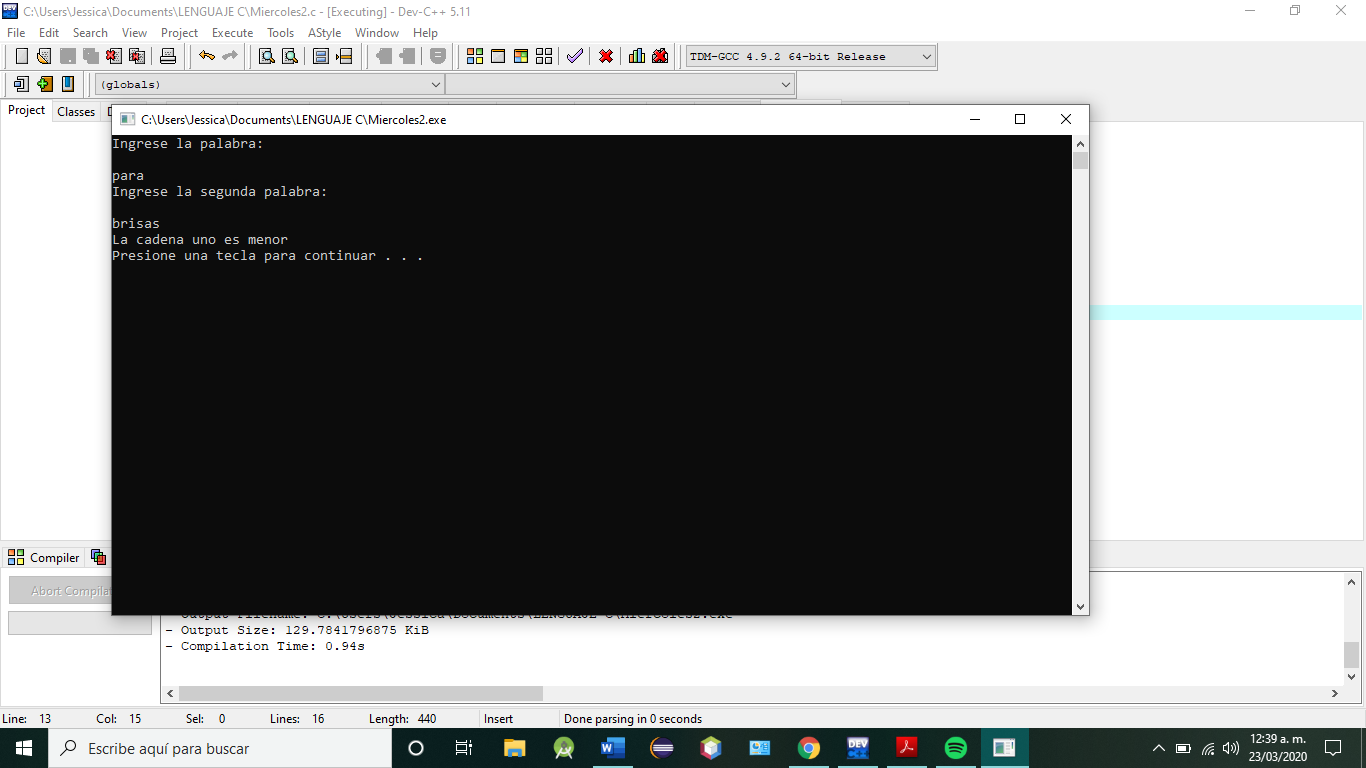
En caso de ser iguales, que imprima la leyenda 'ambas palabras son iguales'.



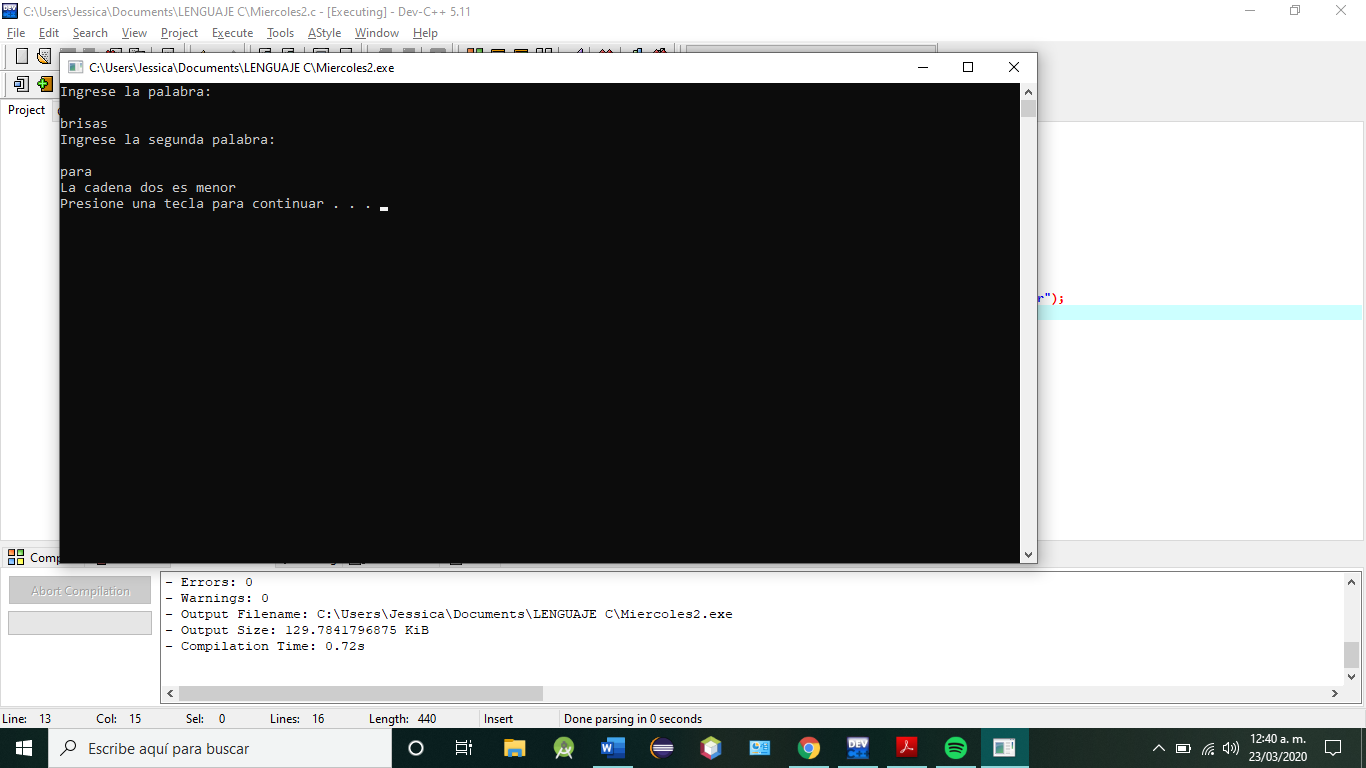
Damos clic en el botón de compilar y ejecutar, o usando la función F11.



Se nos muestra nuestro programa por consola.





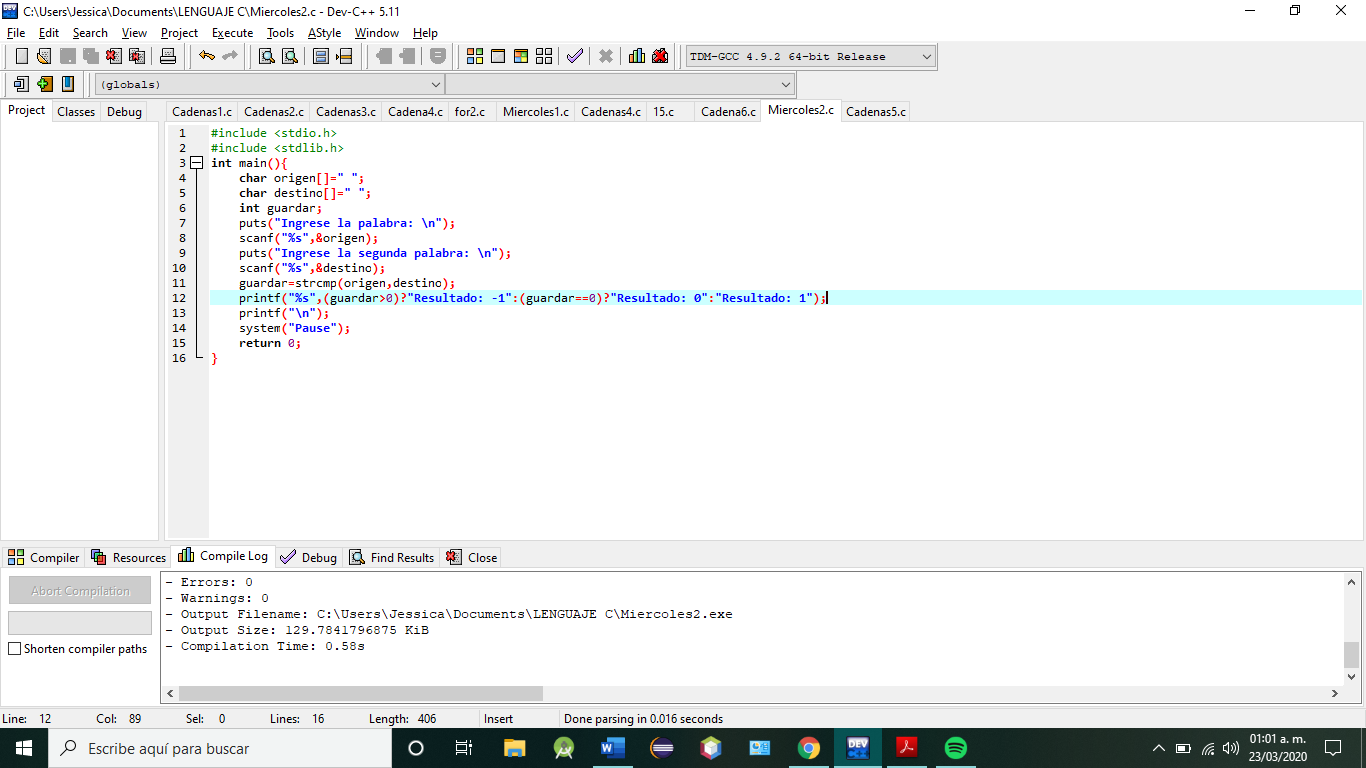


EJERCICIO 67

Compara las dos cadenas y devuelve un 0 si las dos cadenas son iguales.

Un número negativo si <cadena1> es menor que <cadena2>

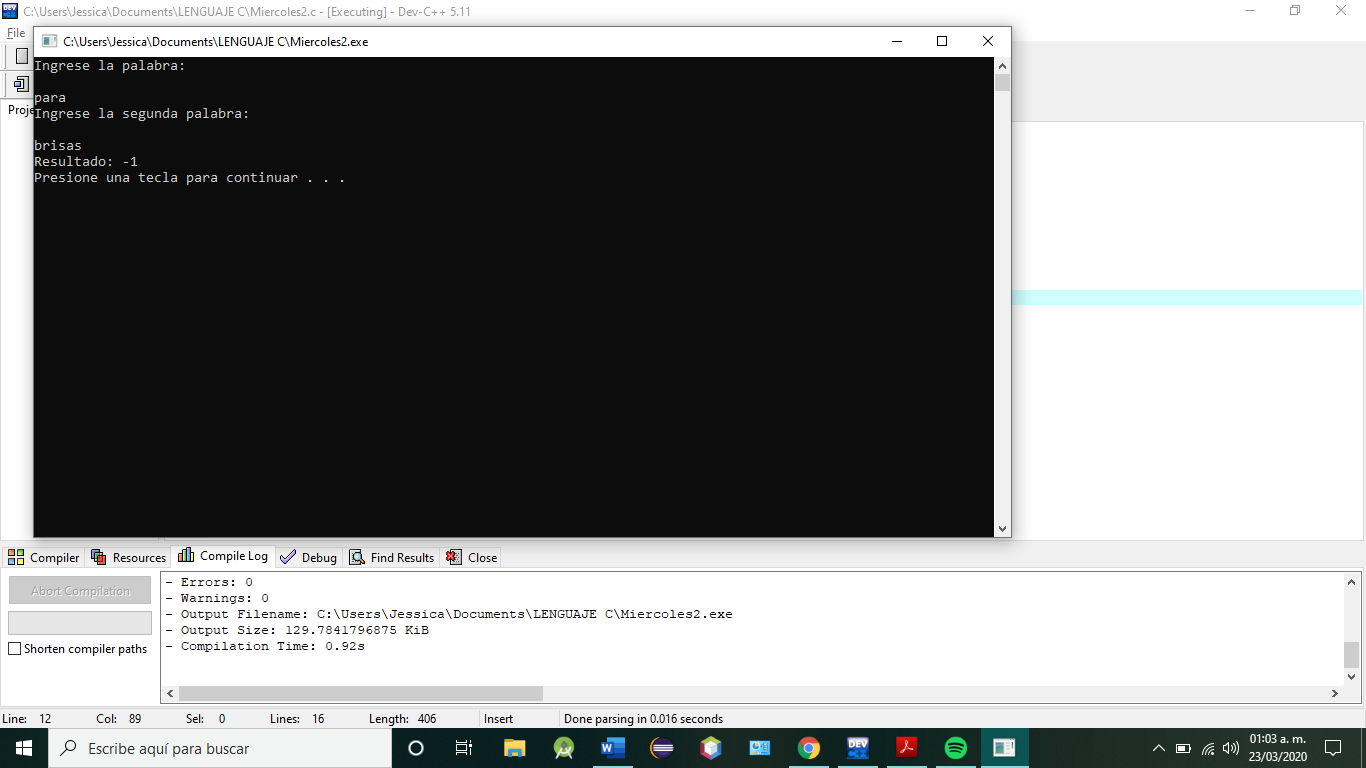
Un número positivo (mayor que cero) si <cadena1> es mayor que <cadena2>.

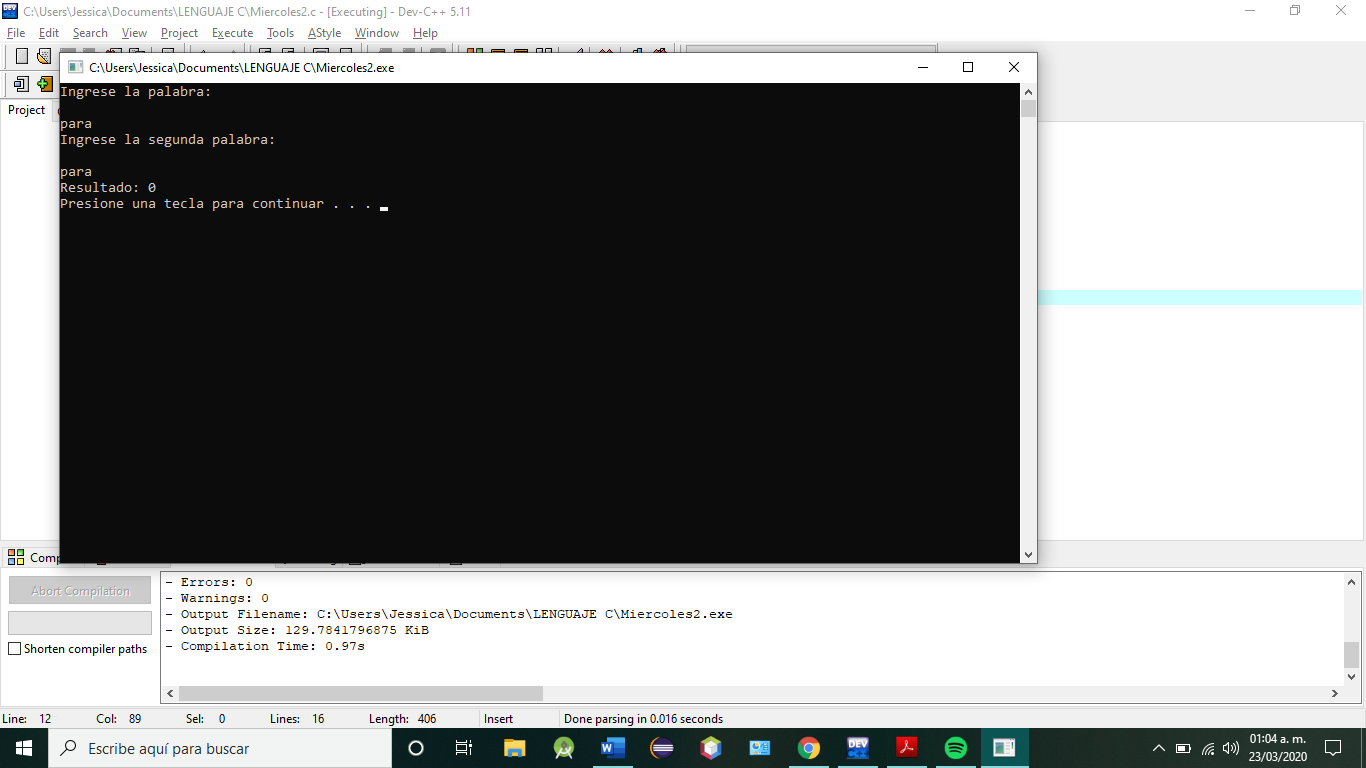


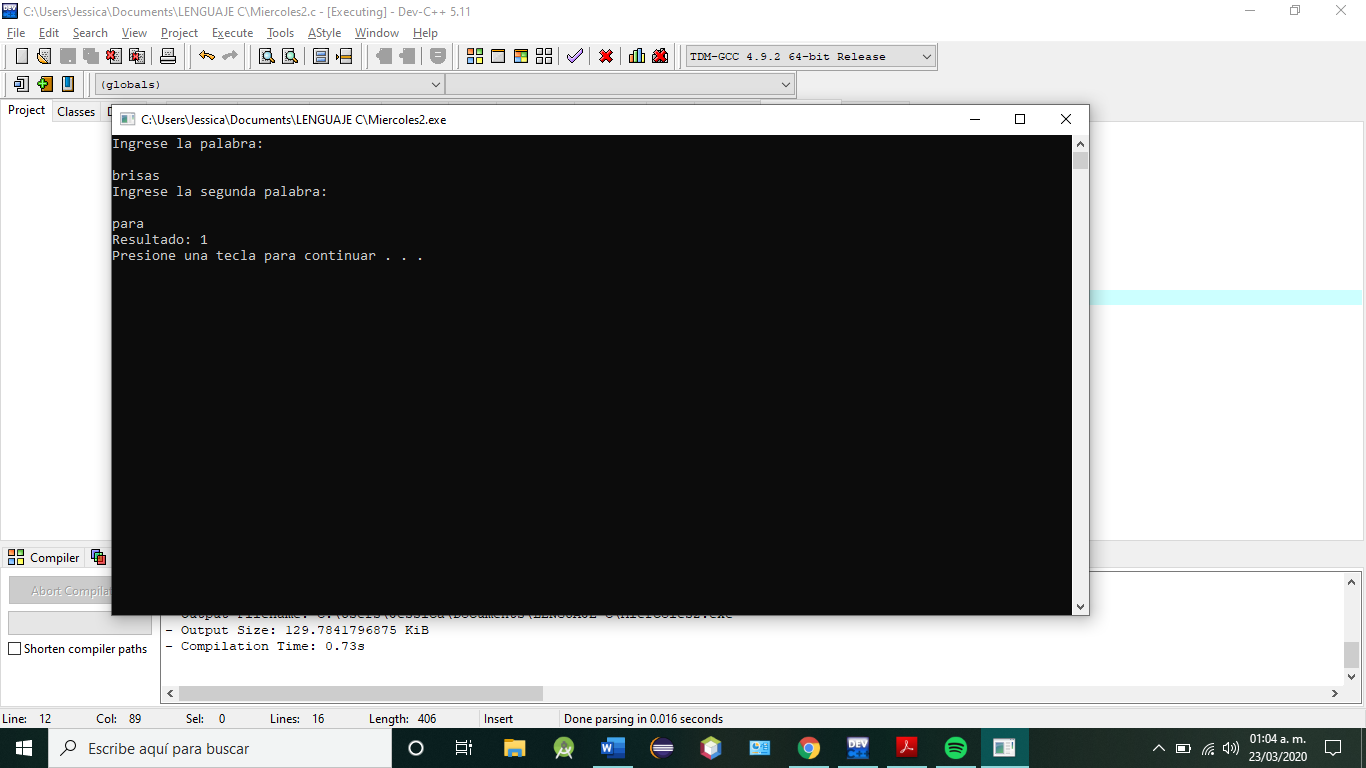
Damos clic en el botón de compilar y ejecutar, o usando la función F11.



Se nos muestra nuestro programa por consola.





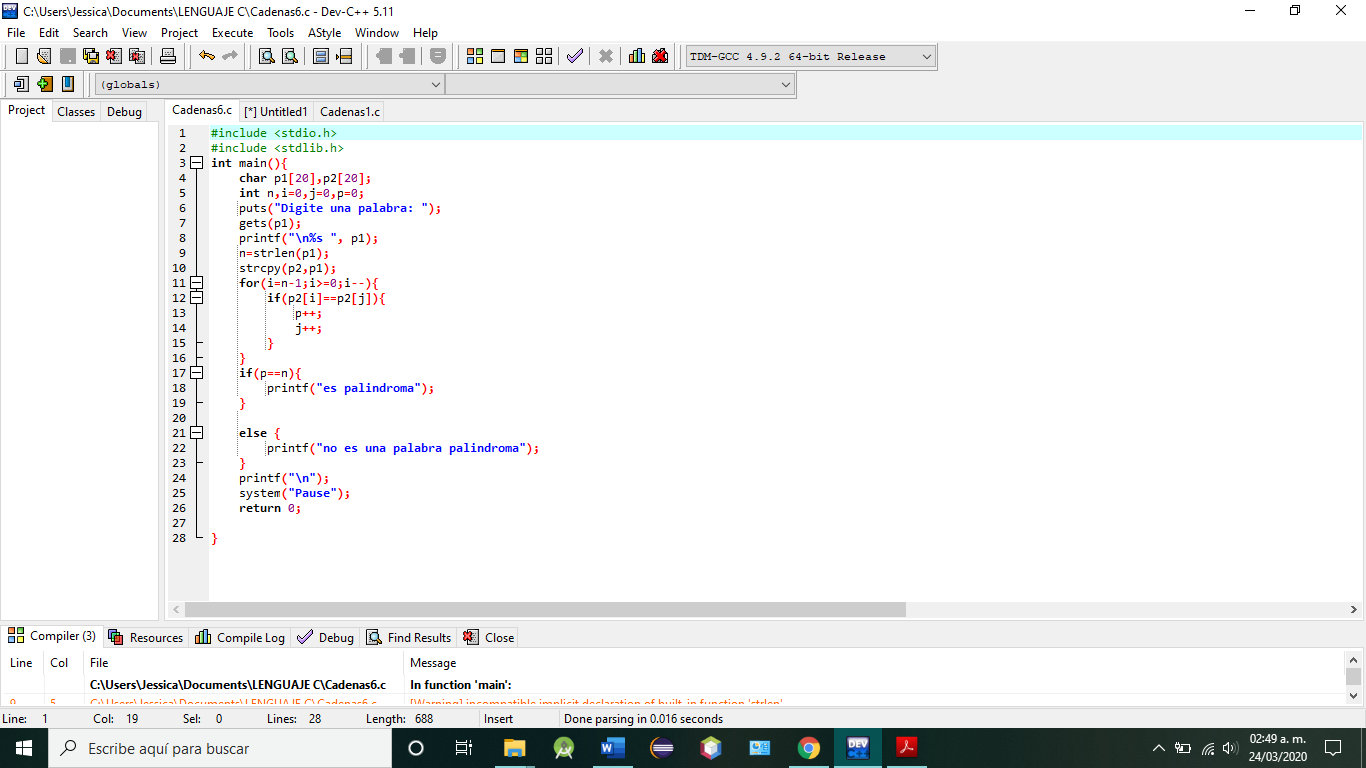


EJERCICIO 68

Crea un programa que detecte una palabra palíndroma.

Los palíndromos son palabras que se leen igual de izquierda a derecha que de derecha a izquierda.

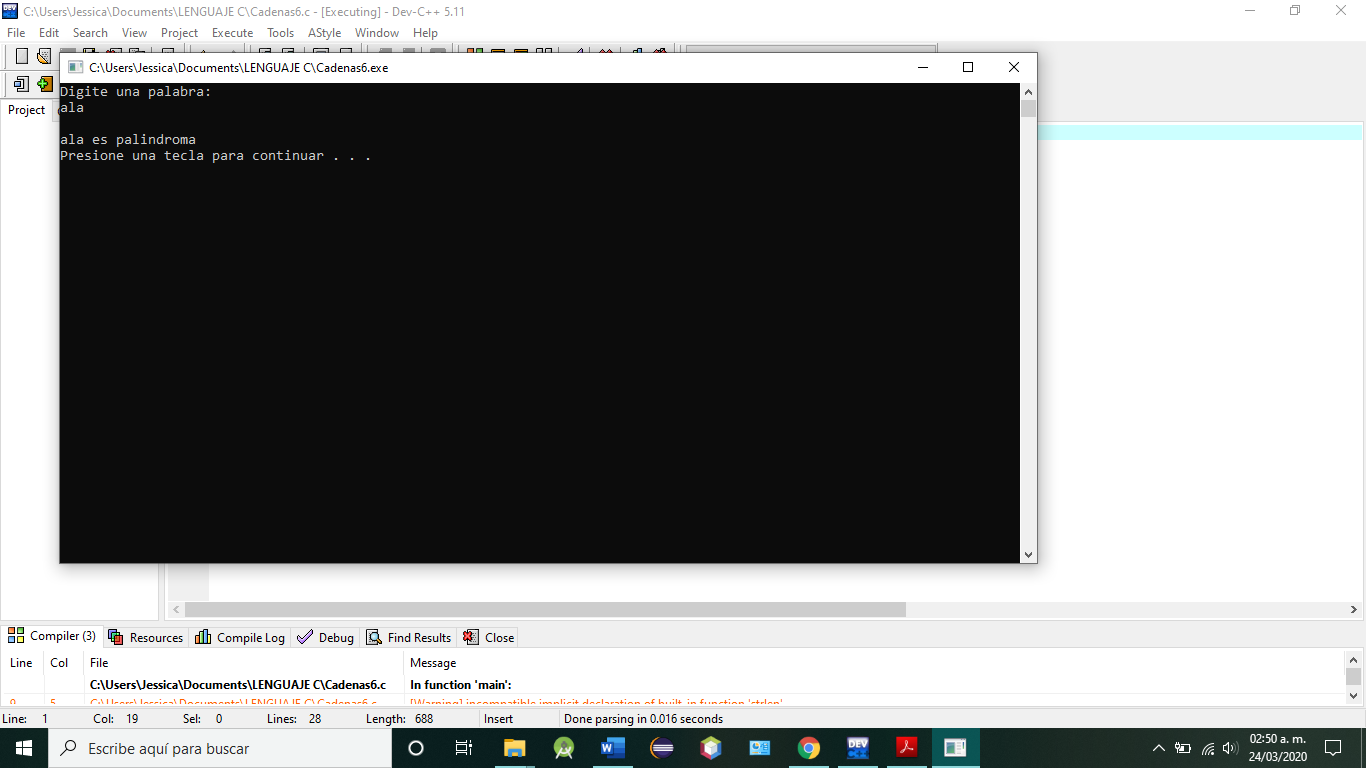
Ejemplo: ala, rotor, salas.

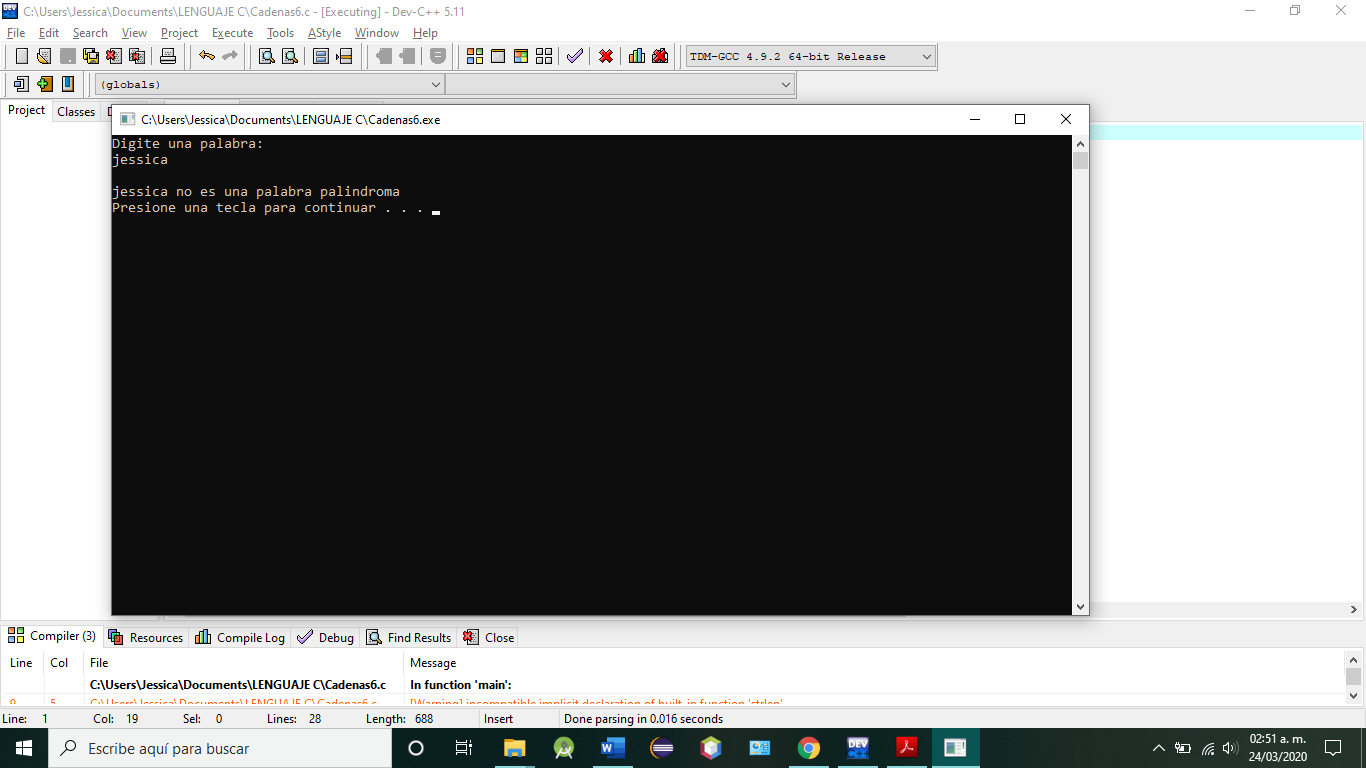


Damos clic en el botón de compilar y ejecutar, o usando la función F11.



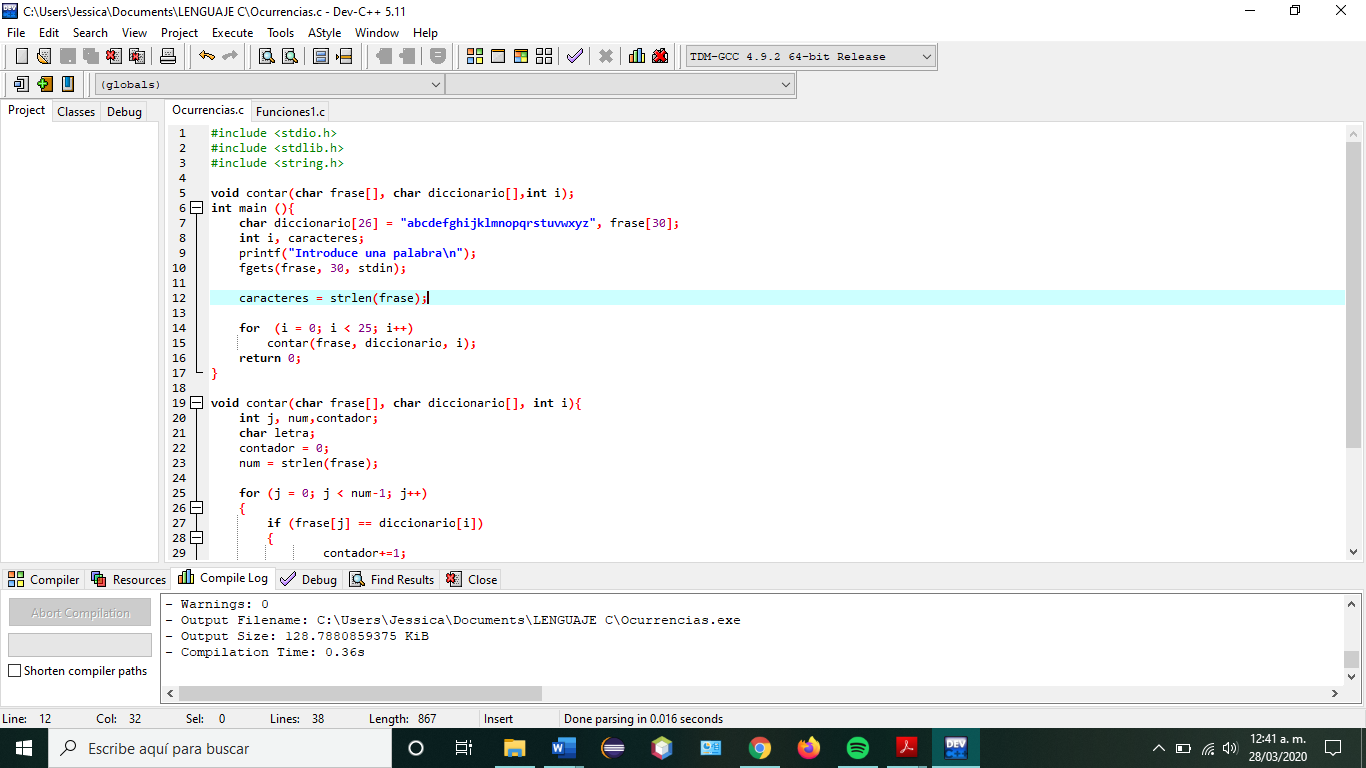
Se nos muestra nuestro programa por consola.

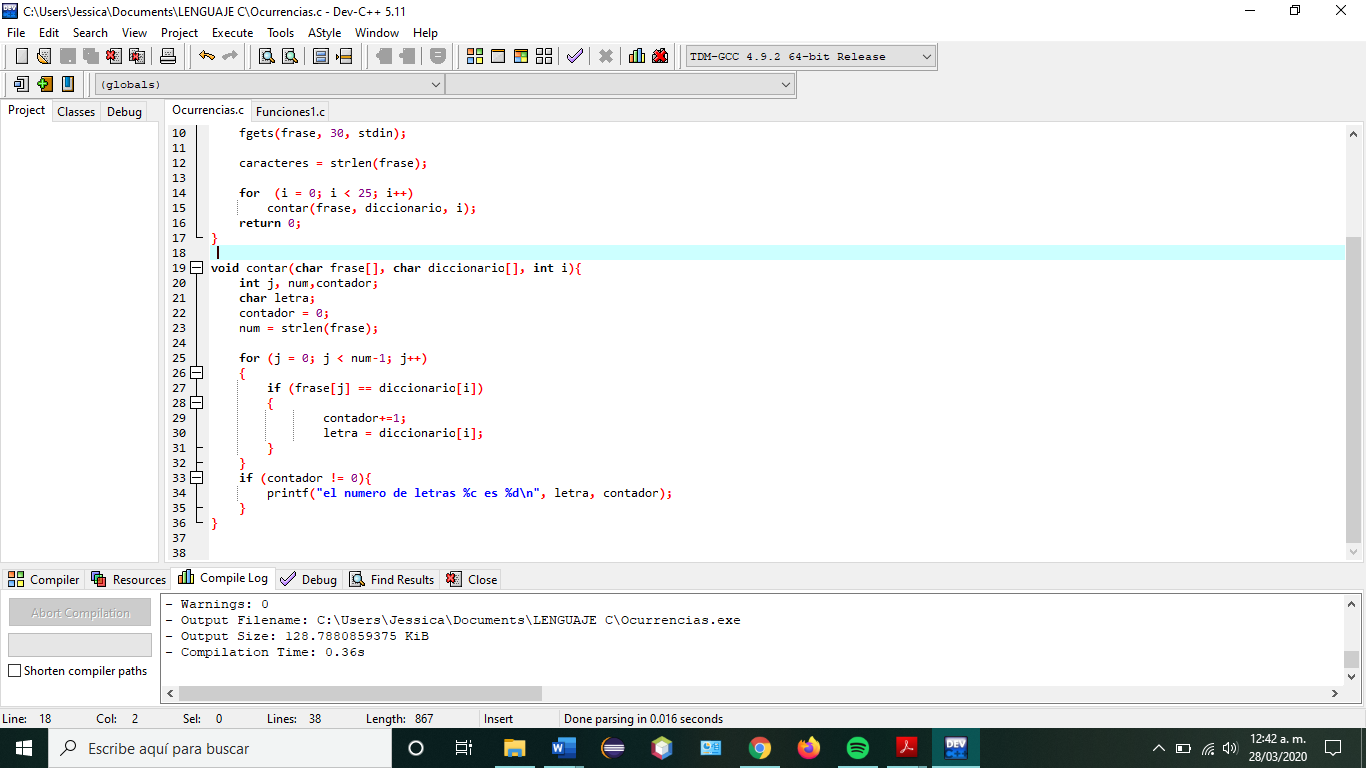




EJERCICIO 69

Crea un programa que cuente cuantas ocurrencias de cada letra contiene una palabra.

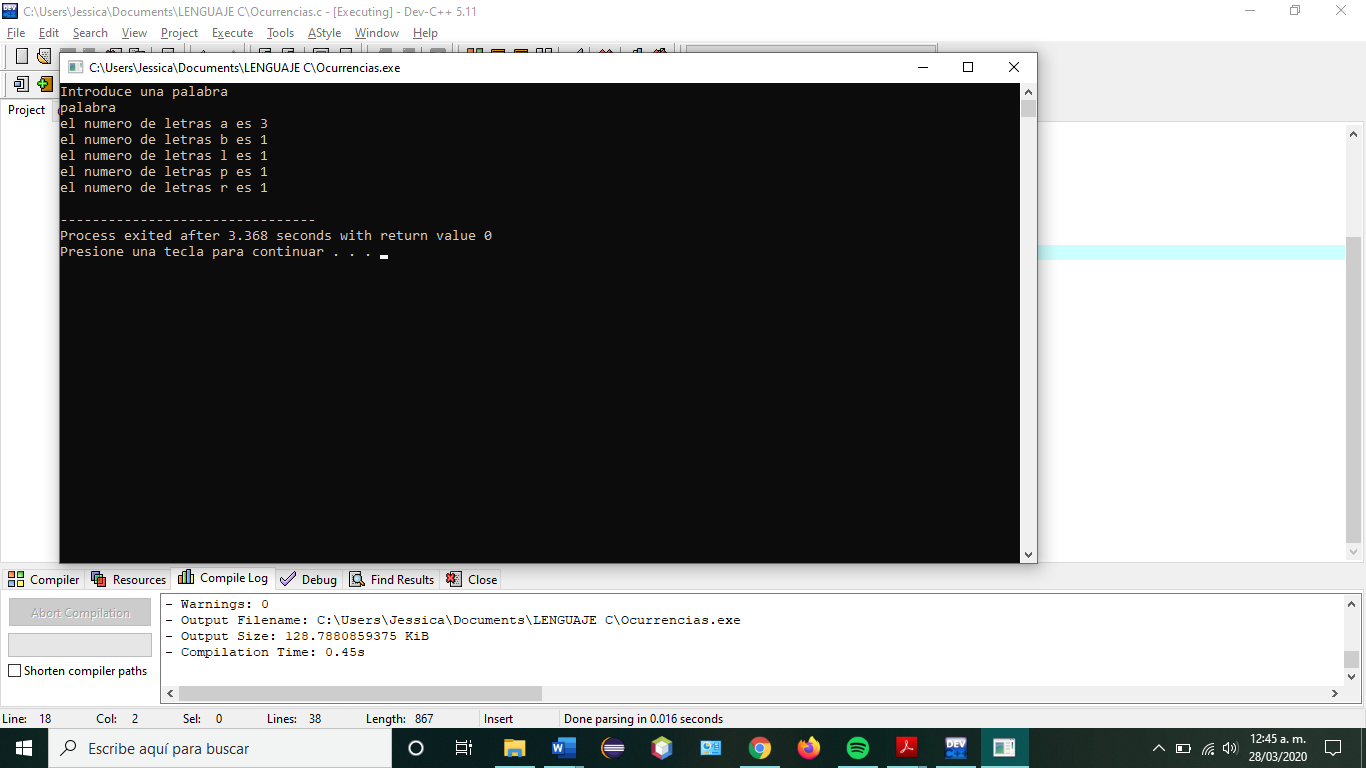




Damos clic en el botón de compilar y ejecutar, o usando la función F11.



Se nos muestra nuestro programa por consola.



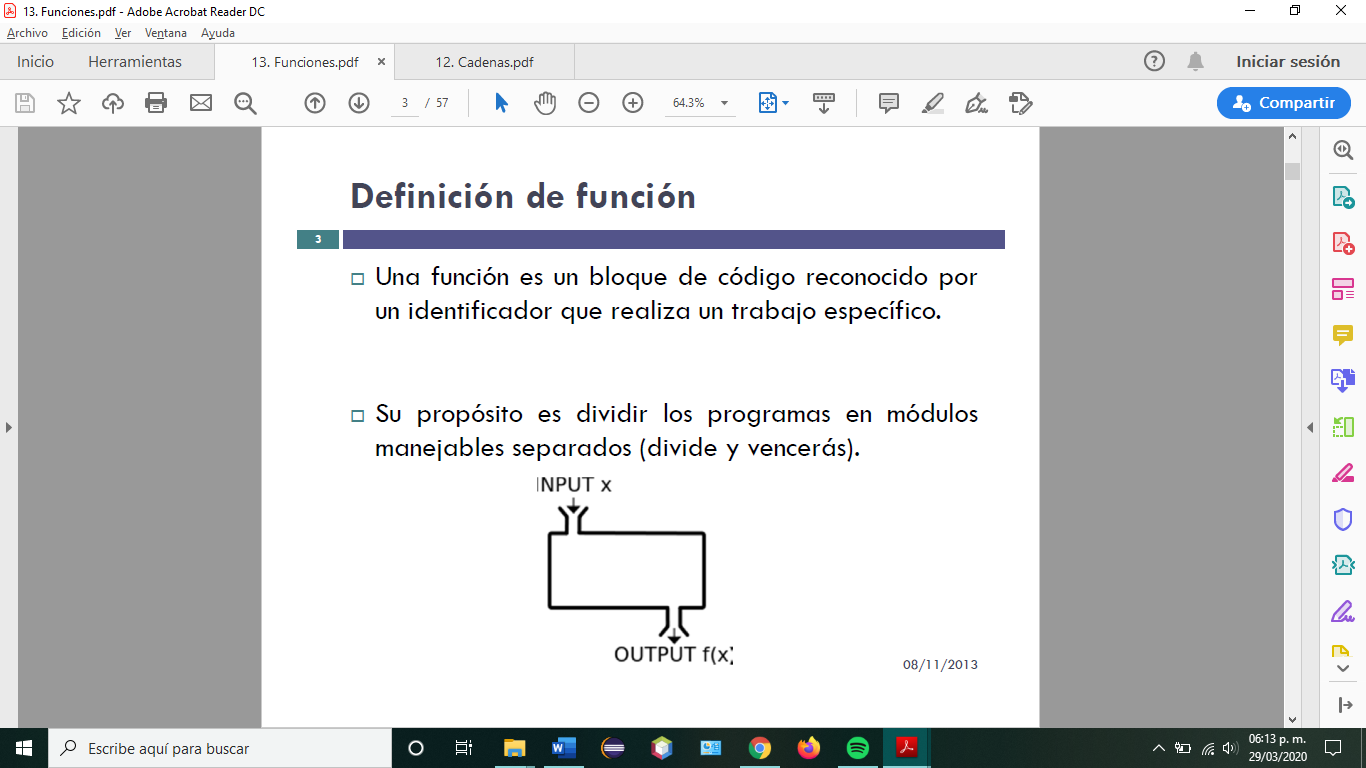
EJERCICIO 70

Ejercicio con funciones.

Matemáticamente una función es una operación que toma uno o más valores llamados argumentos y produce un valor llamado resultado.

Una función es un bloque de código reconocido por un identificador que realiza un trabajo específico.

Su propósito es dividir los programas en módulos manejables separados (divide y vencerás).



Ventajas

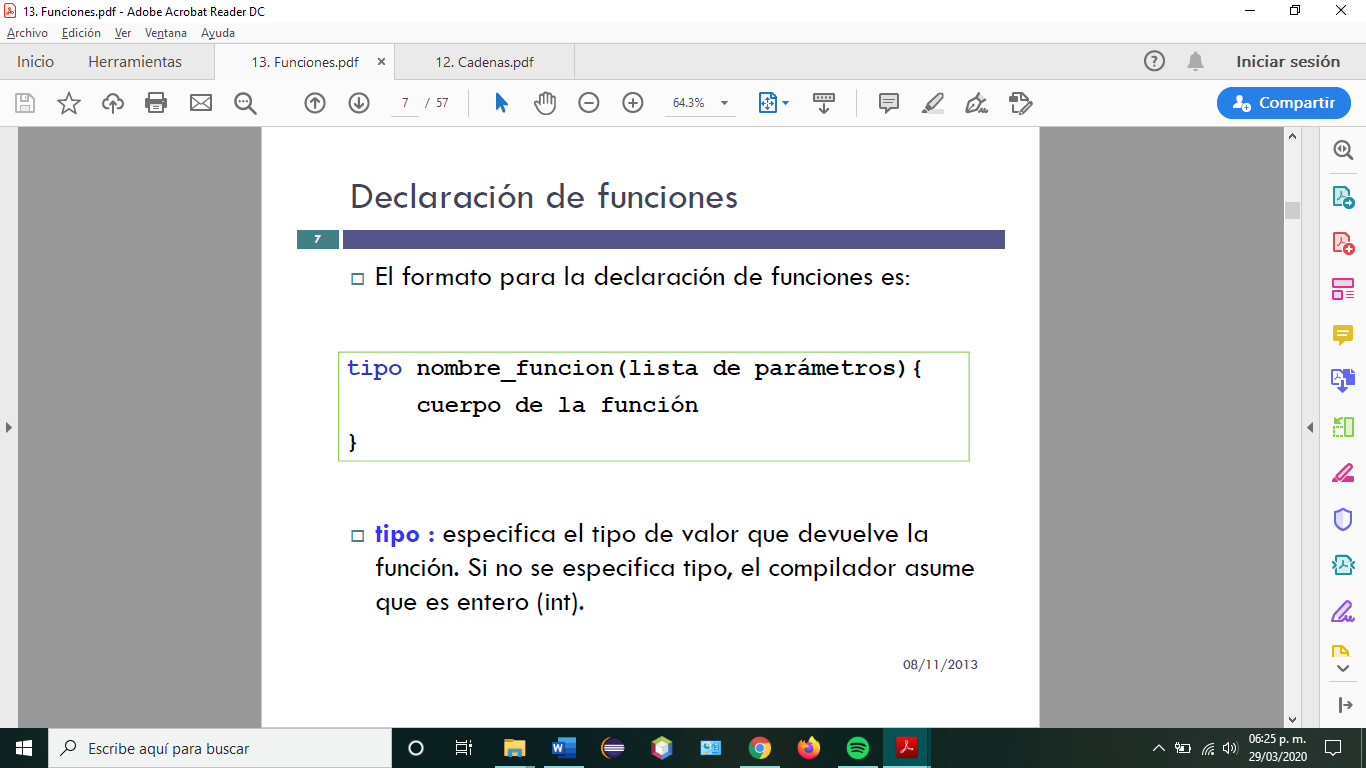
* Facilita el diseño descendente.
* Los procedimientos dentro de ellas se pueden ejecutar varias veces.
* Facilita la división de tareas.
* Se pueden probar individualmente
* Con funciones apropiadamente diseñadas, es posible ignorar como se realiza una tarea, sabiendo qué es lo que hacen.

Modo de uso

* Funciones diseñadas para realizar operaciones a partir de sus argumentos y devolver un valor basado en sus cálculos.
* Funciones que no reciben argumentos, realizan un proceso y devuelven un valor.
* Funciones que no tienen argumentos ni valor de retorno explícito, operan sobre el entorno de variables globales o atributos del sistema operativo.

Declaración de funciones

El formato para la declaración de funciones es:



tipo: especifica el tipo de valor que devuelve la función. Si no se especifica tipo, el compilador asume que es entero (int).

Declaración de funciones 2



lista de parámetros: es la lista de nombres de variables separados por comas con sus tipos asociados que reciben los valores de los argumentos actuales de la llamada a la función.

Entre llaves se encuentra el cuerpo de la función.

Declaración de funciones 3

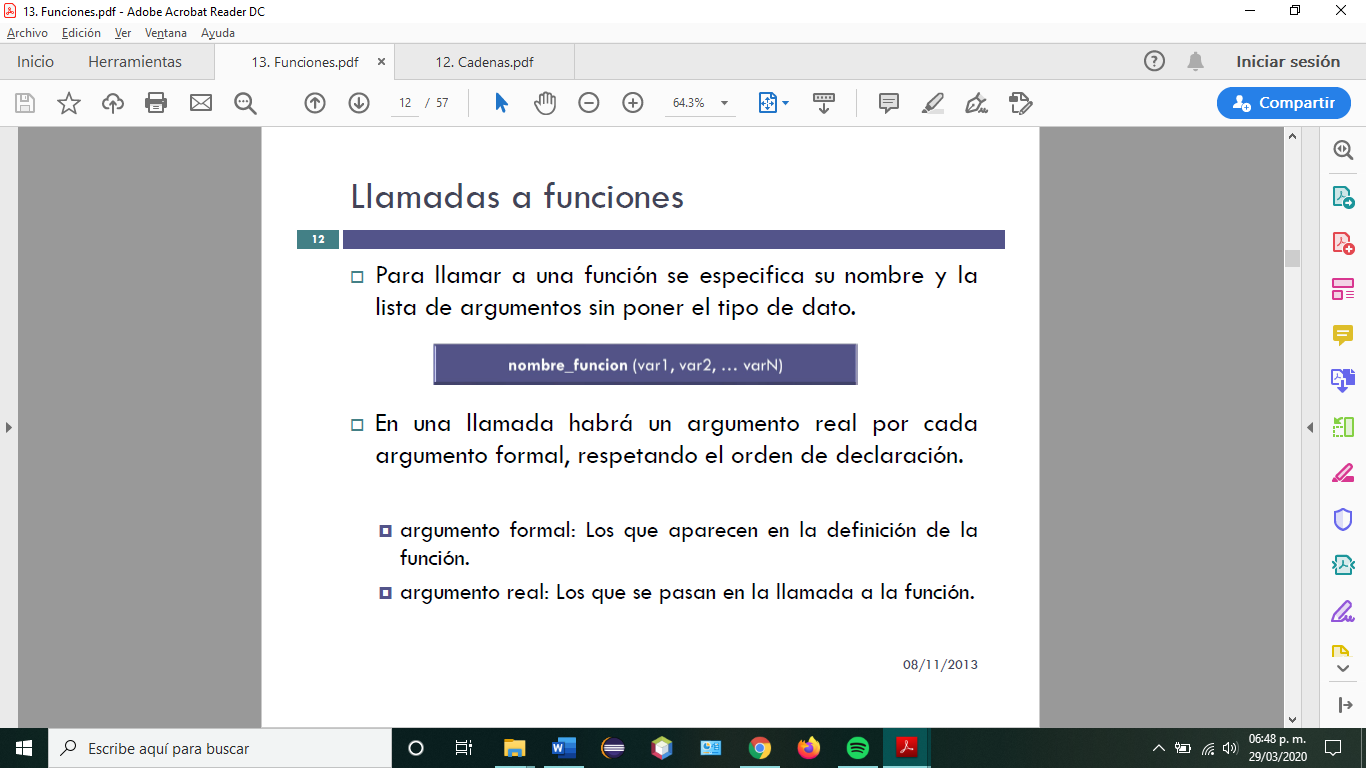
* Forza la salida inmediata de la función en que se encuentra.
* Una función puede retornar valor sólo cuando el tipo de retorno no es void.
* Devuelve un valor a la función que realizó la llamada.
* return (expresion);

Declaración de funciones 4

* Tradicionalmente en C se declaran como prototipos al inicio del programa. Después se declara la función main, y después se hace la declaración formal de las funciones.
* También pueden declararse las funciones al inicio del programa y después declarar la función main sin declarar prototipo.

Llamadas a una función

Para llamar a una función se especifica su nombre y la lista de argumentos sin poner el tipo de dato.



En una llamada habrá un argumento real por cada argumento formal, respetando el orden de declaración.

* argumento formal: Los que aparecen en la definición de la función.
* argumento real: Los que se pasan en la llamada a la función.

Paso de parámetros por valor

Se hace una copia del valor del argumento en el parámetro formal.

* La función opera internamente con estos últimos.
* Los parámetros formales se crean al entrar a la función y se destruyen al salir de ella, cualquier cambio realizado por la función en los parámetros formales no tienen ningún efecto sobre los argumentos.

Variables locales y globales

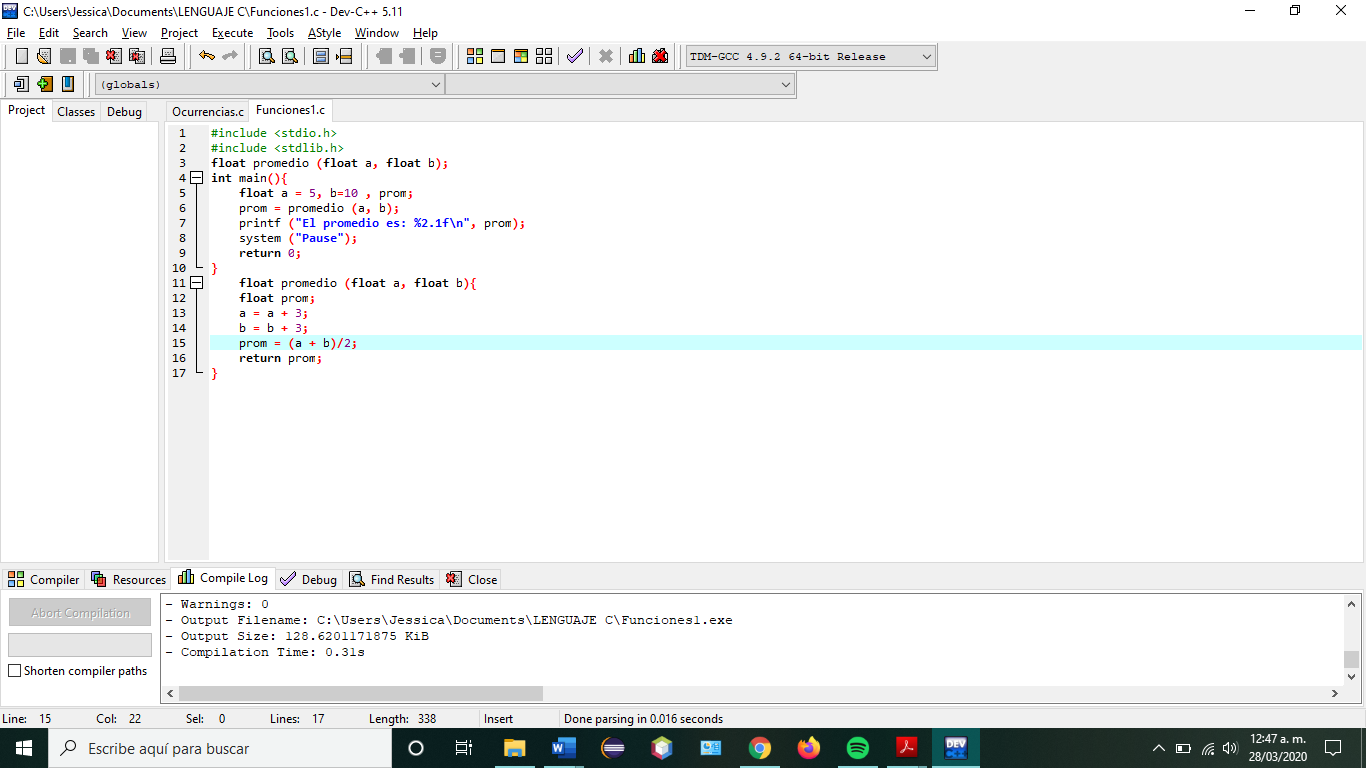
Variables Locales:

* Se declaran dentro de la función y sólo están disponibles durante su ejecución.
* Se crean cuando se entra en ejecución una función y se destruyen cuando se termina.

Variables globales:

* Se declaran fuera de las funciones. Pueden ser utilizadas por todas las funciones.
* Existen durante toda la vida del programa.

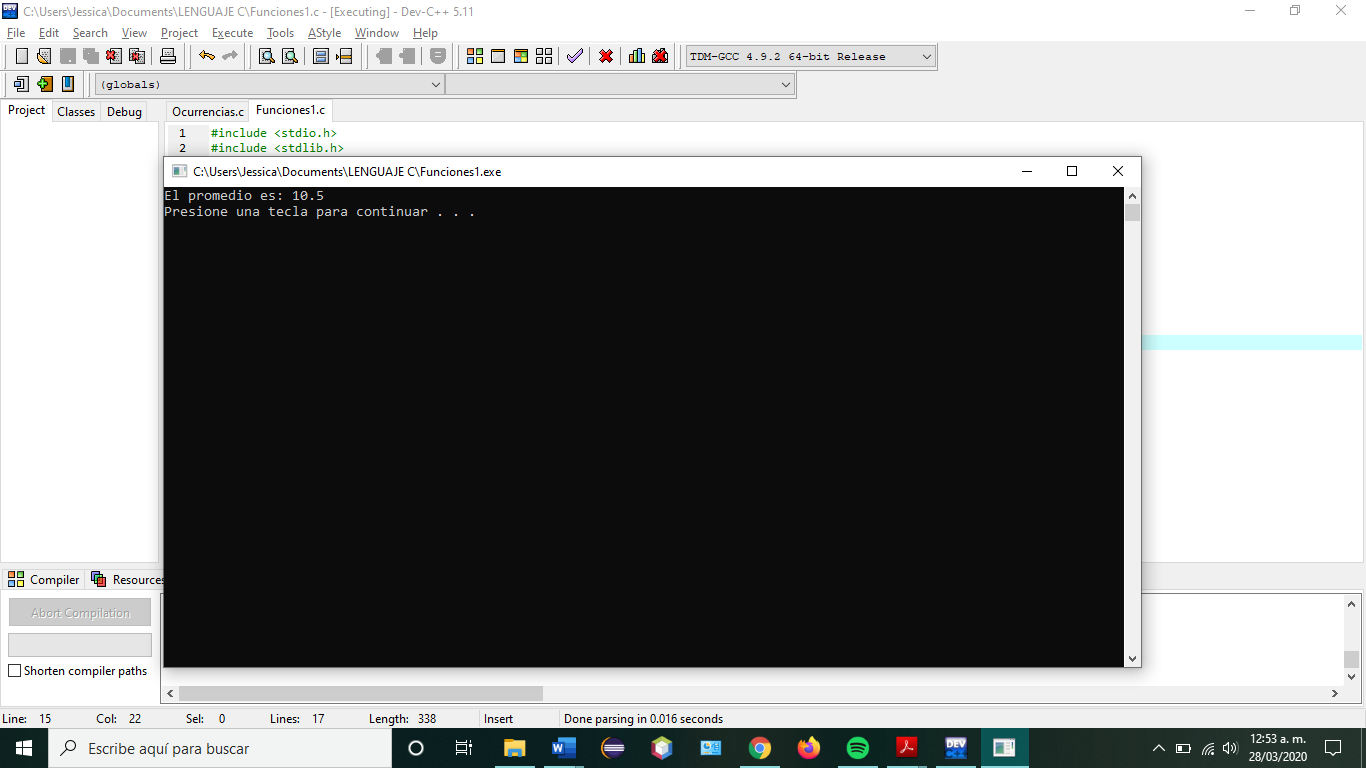
Ejemplos:



Damos clic en el botón de compilar y ejecutar, o usando la función F11.

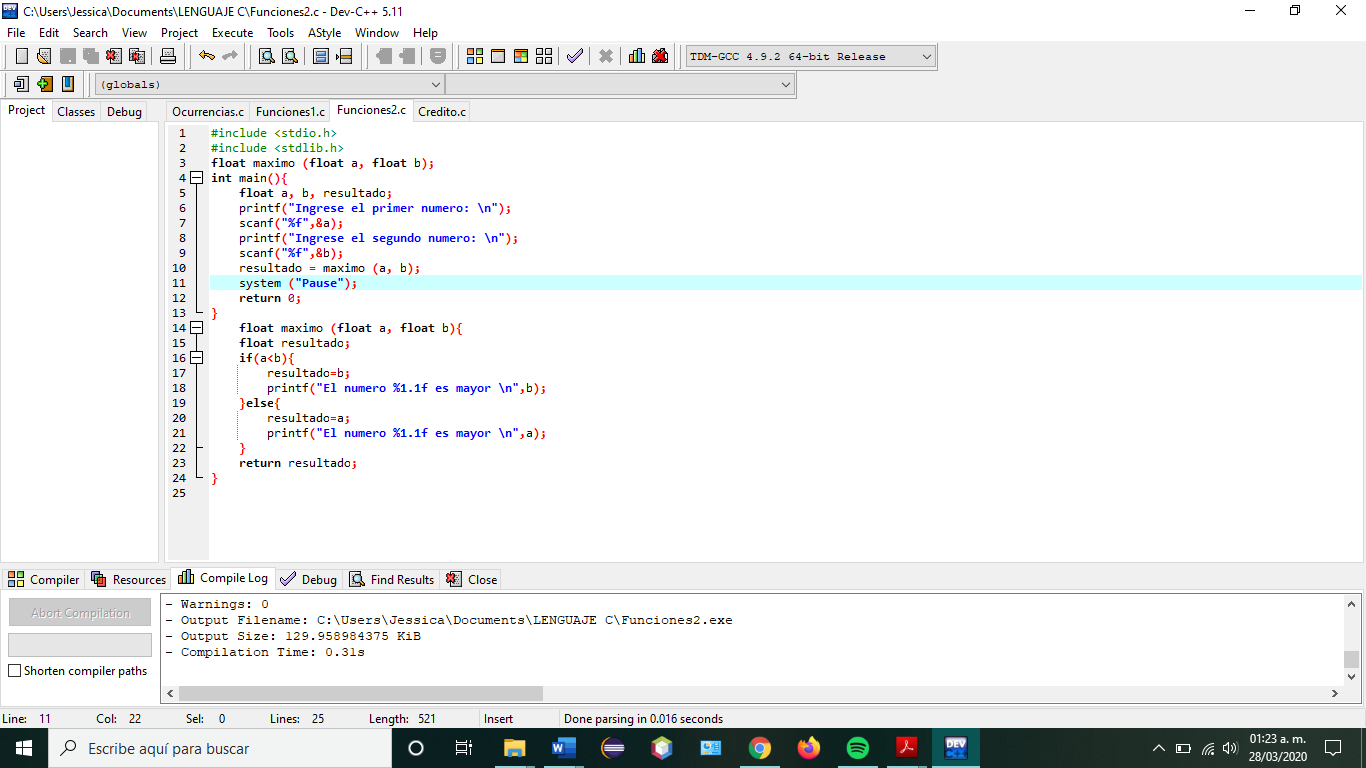


Se nos muestra nuestro programa por consola.



EJERCICIO 71

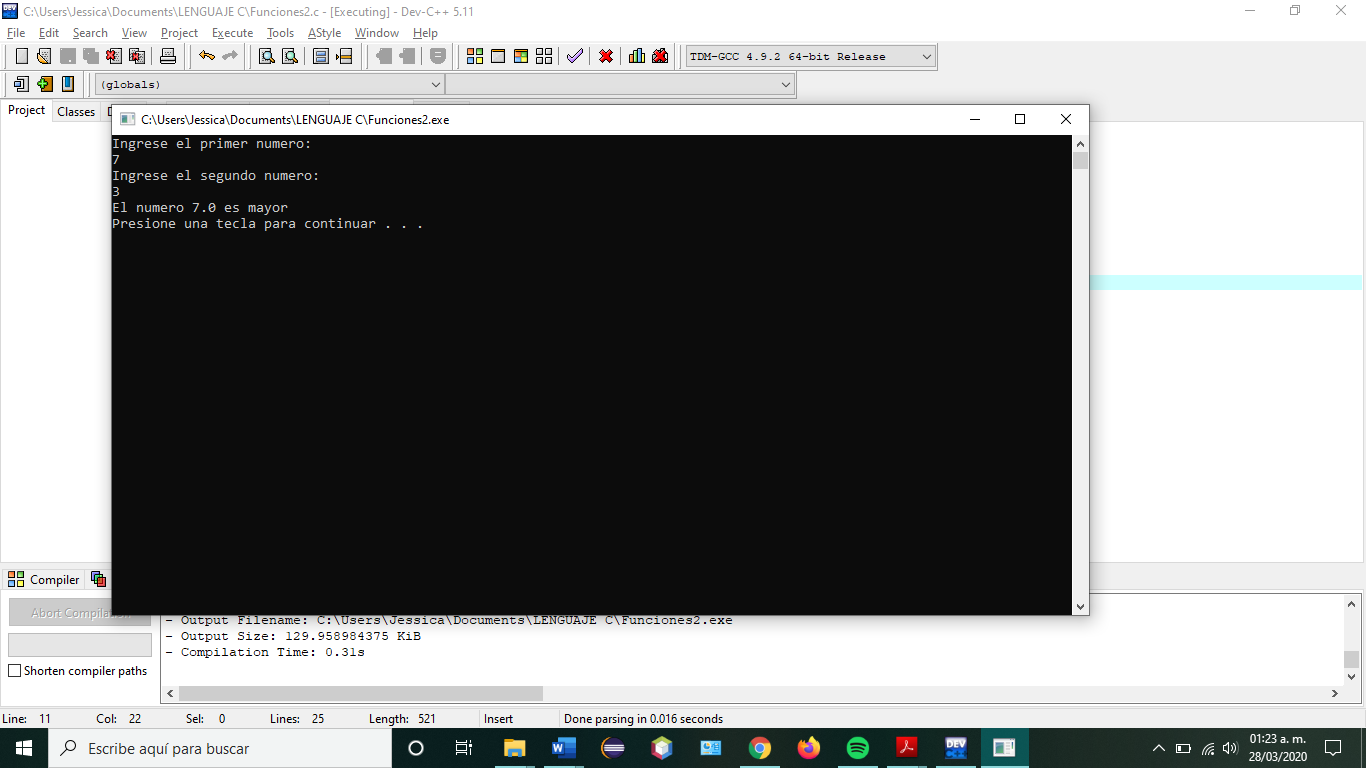
Escribir una función que se llame máximo que reciba dos números por parámetros y que regrese el mayor de ellos.

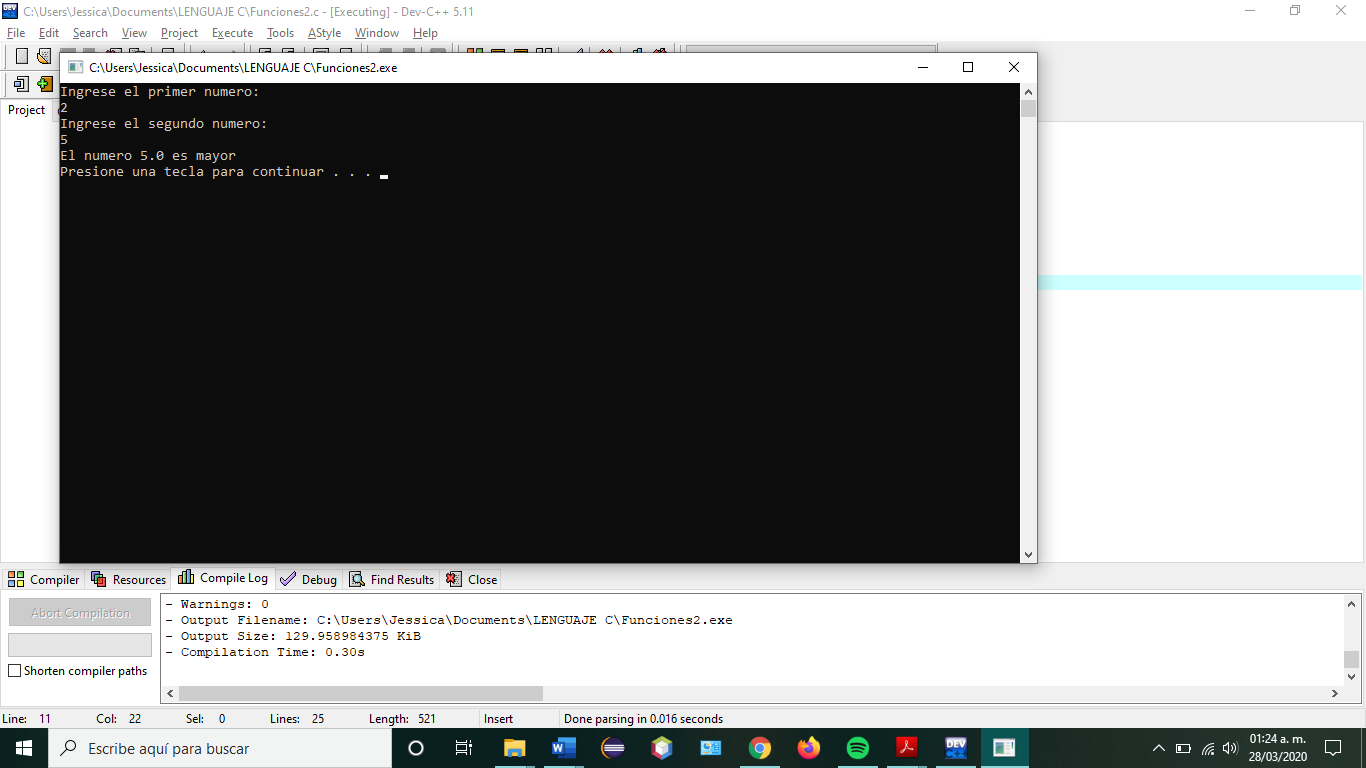


Damos clic en el botón de compilar y ejecutar, o usando la función F11.



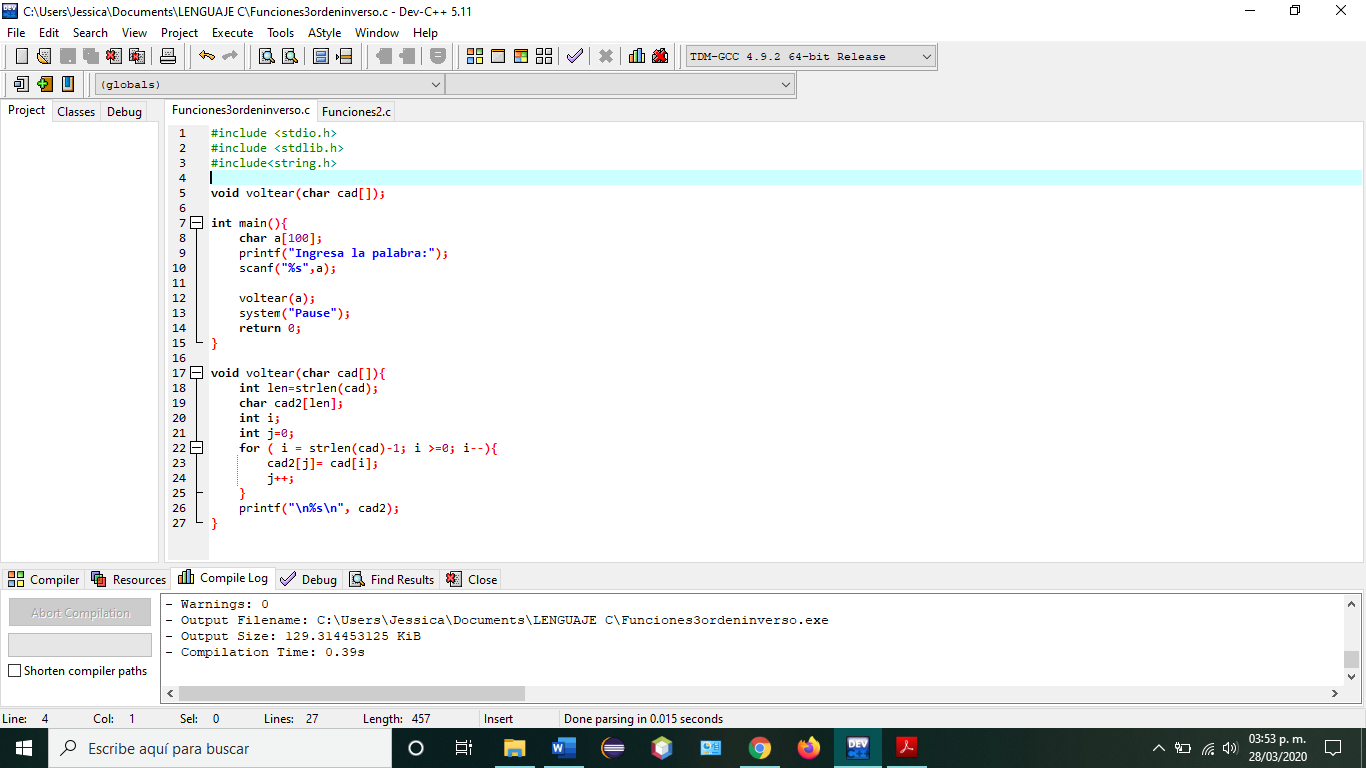
Se nos muestra nuestro programa por consola.





EJERCICIO 72

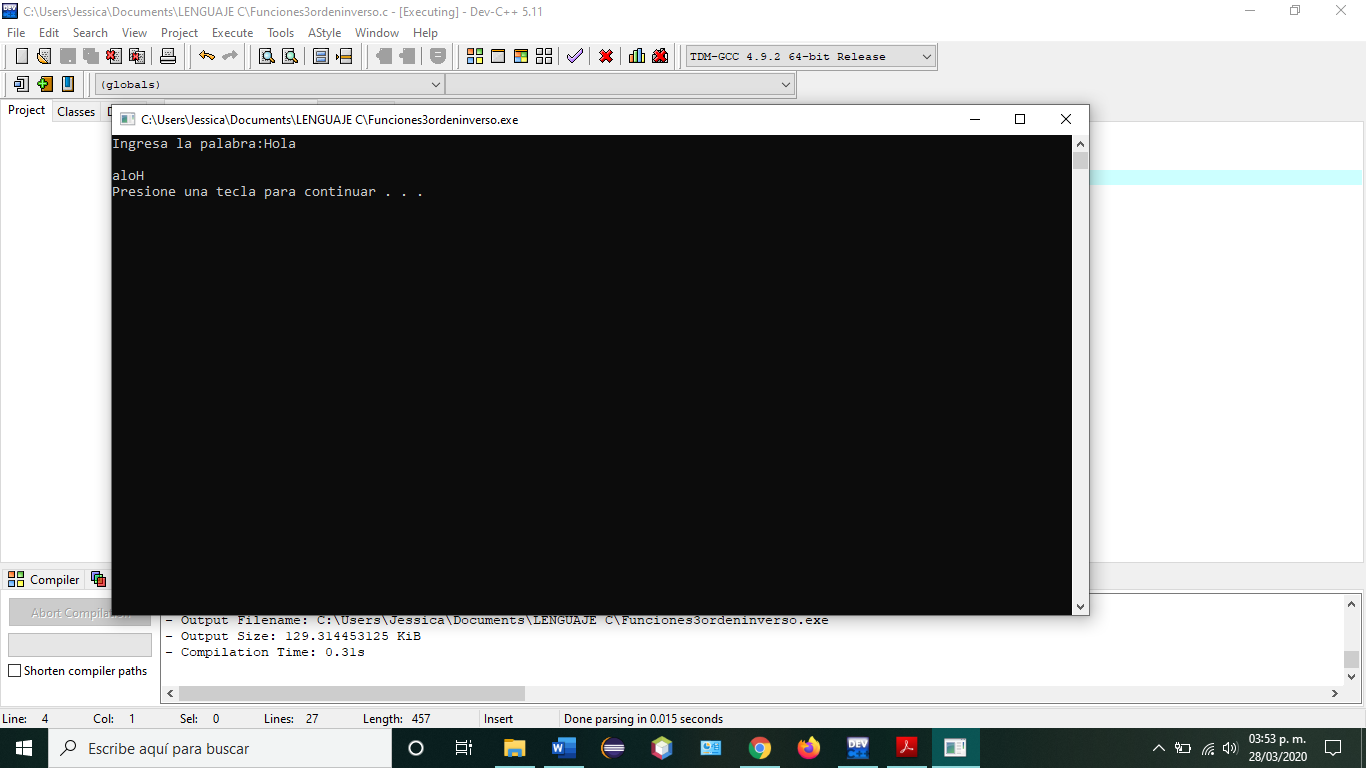
Escribir una función que reciba caracteres del teclado hasta recibir un espacio o un salto de línea (enter) y a continuación mostrar todos los caracteres en orden inverso.



Damos clic en el botón de compilar y ejecutar, o usando la función F11.

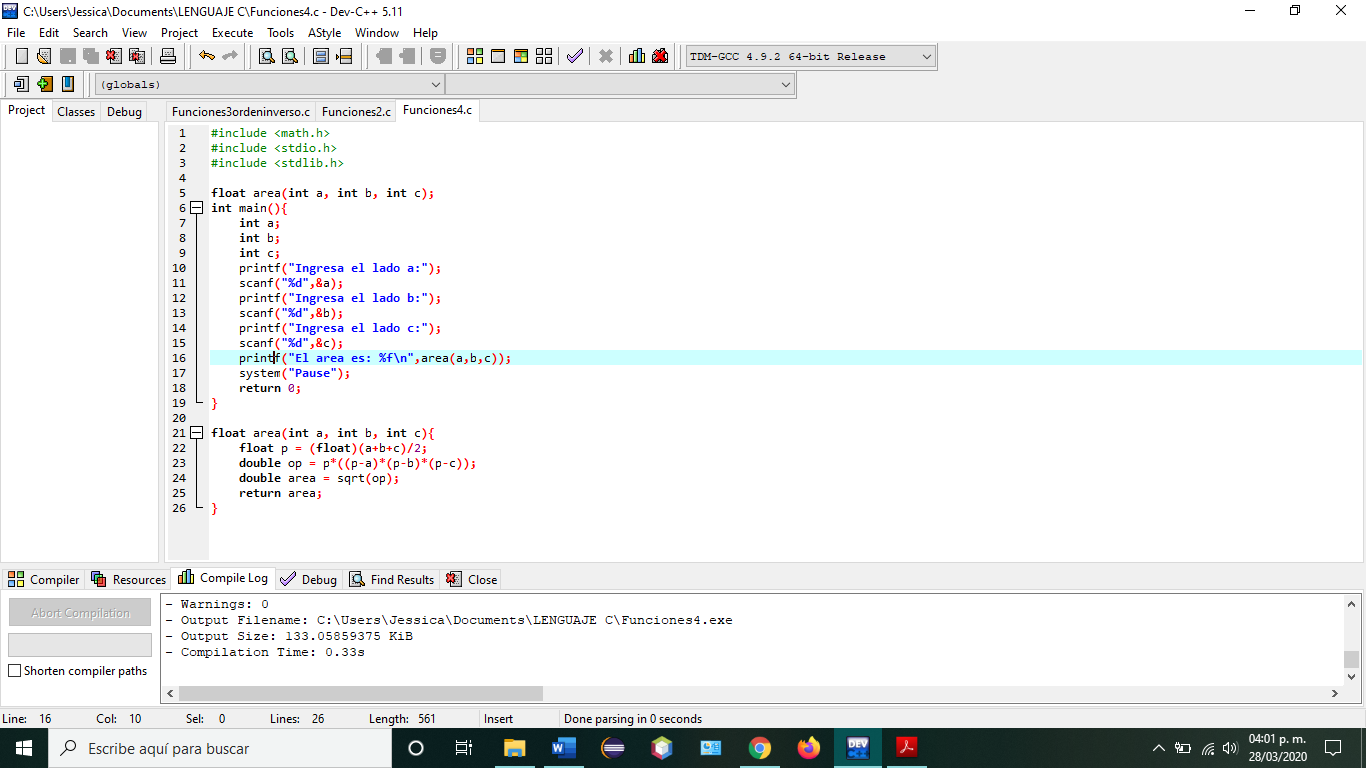


Se nos muestra nuestro programa por consola.



EJERCICIO 73

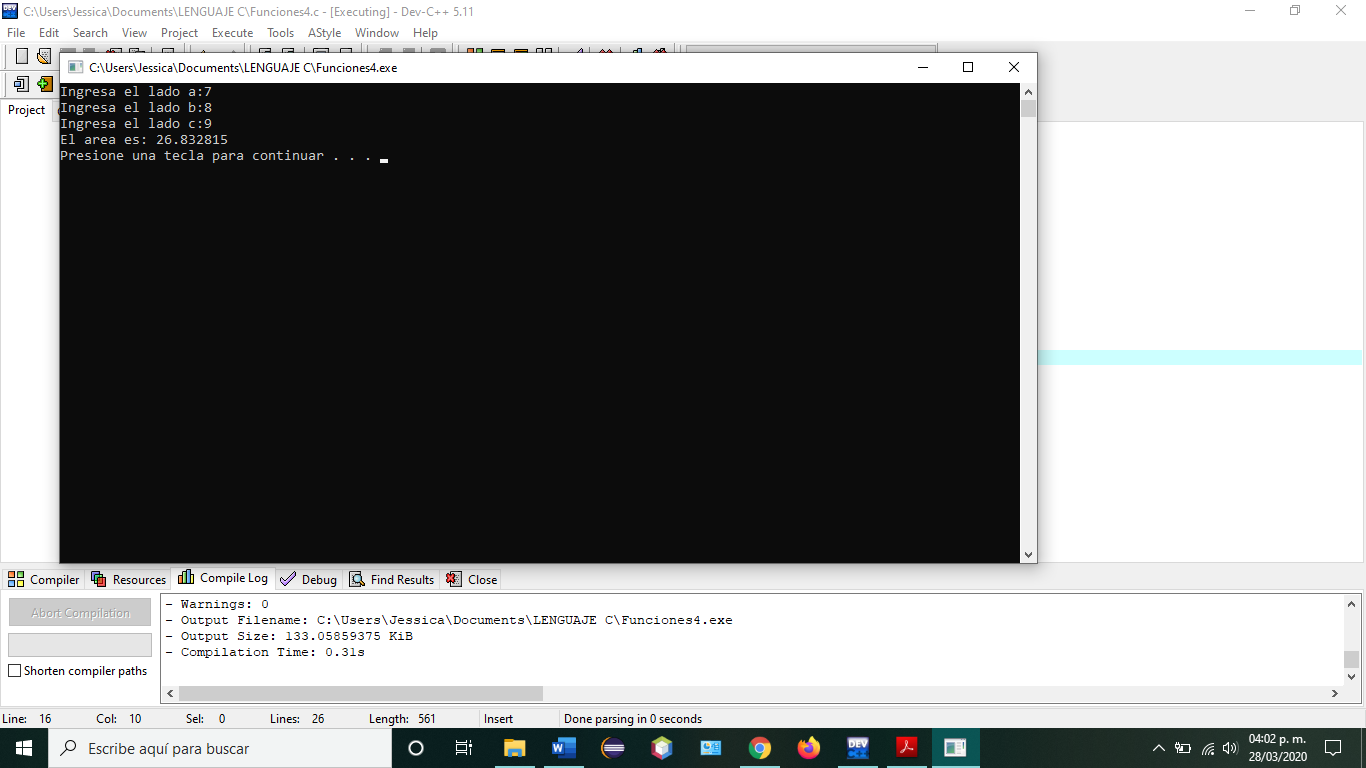
Escribir una función que tome como parámetros las longitudes de los tres lados de un triángulo (a, b, c) y devuelva el área del triángulo.



Damos clic en el botón de compilar y ejecutar, o usando la función F11.



Se nos muestra nuestro programa por consola.



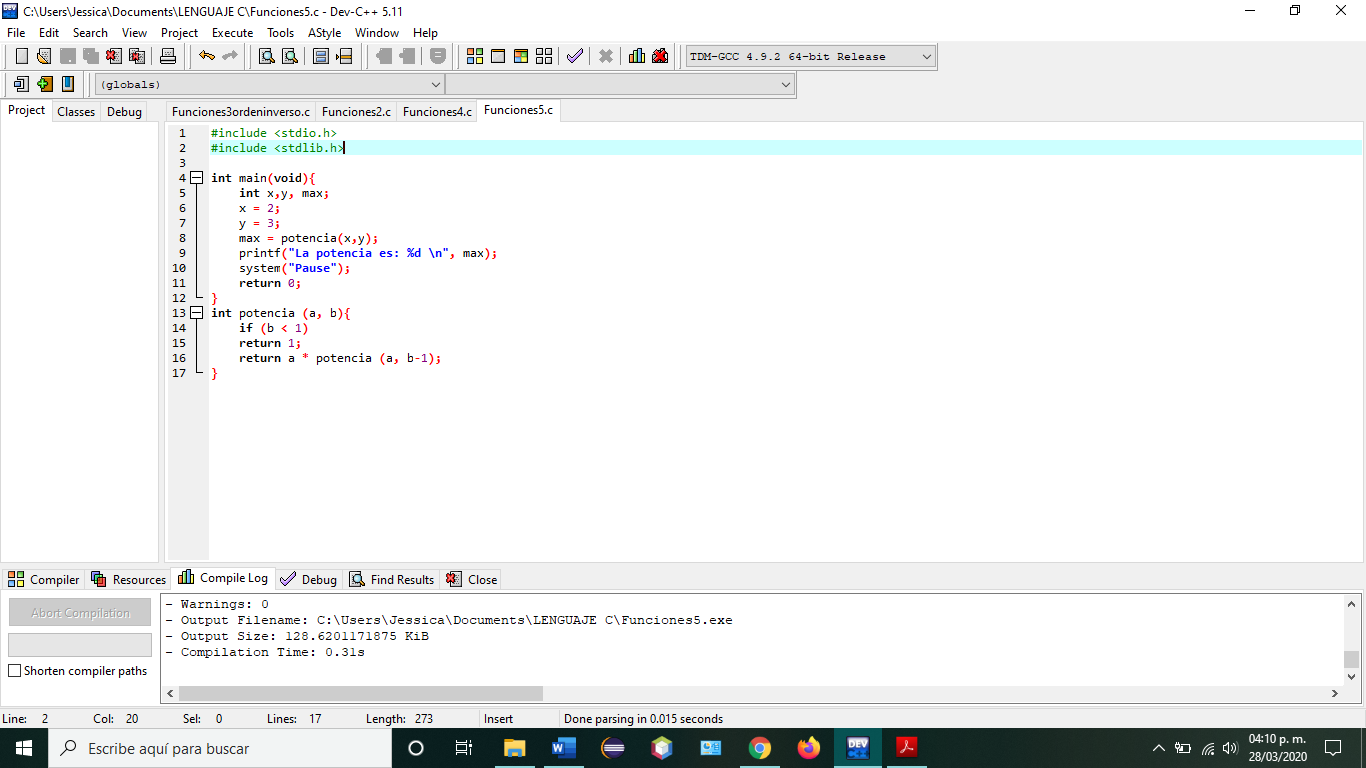
EJERCICIO 74

Funciones recursivas

Se llaman funciones recursivas a aquellas que se llaman a su mismas de forma repetida hasta que se cumpla alguna condición.

Cada llamada implica el almacenamiento de variables de estado y otros parámetros

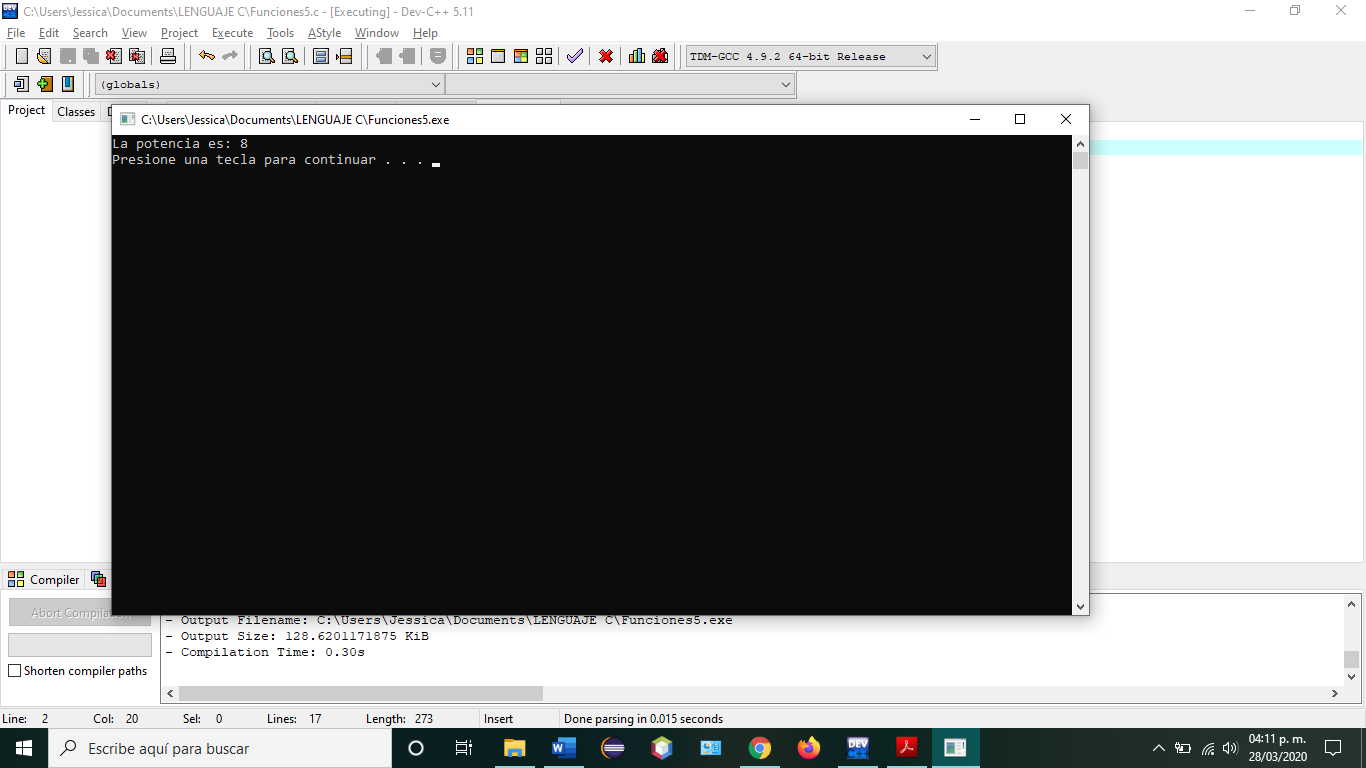
Calcular la potencia de forma recursiva



Damos clic en el botón de compilar y ejecutar, o usando la función F11.

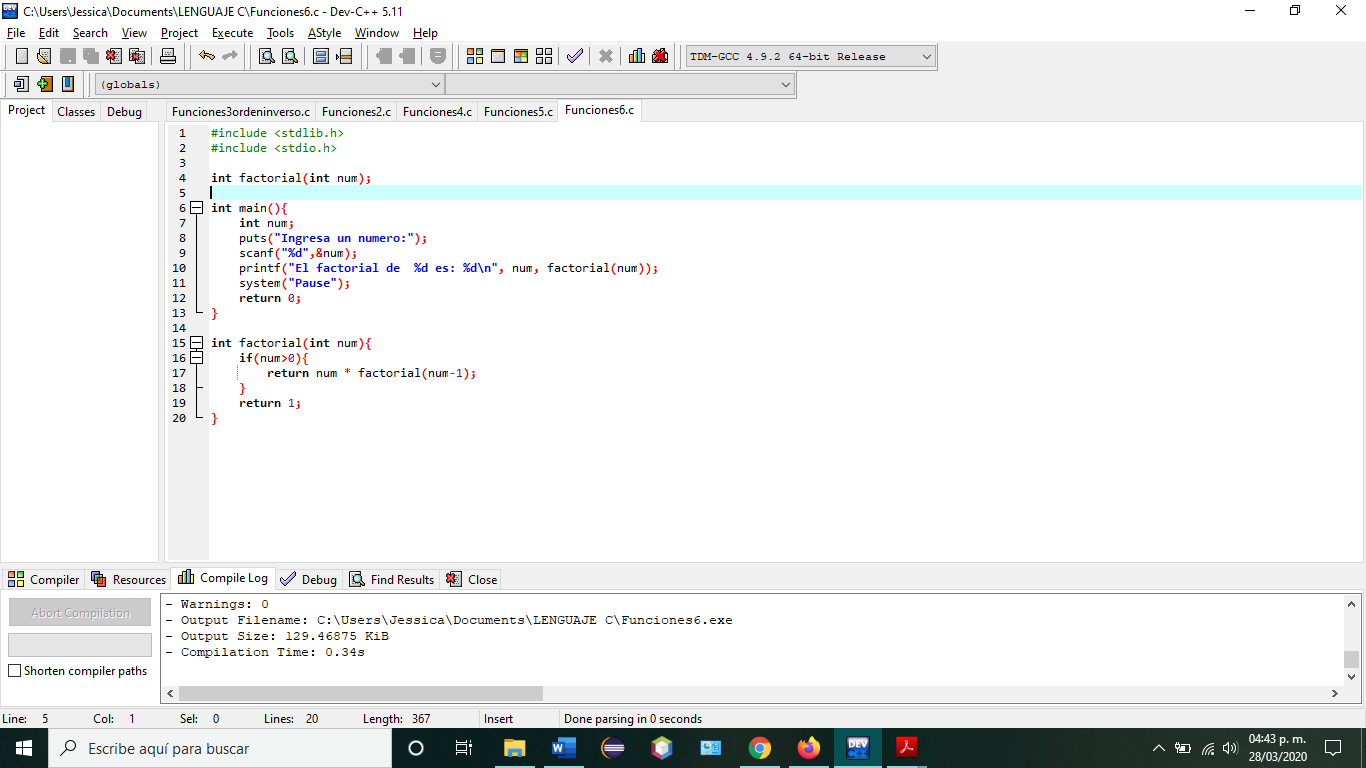


Se nos muestra nuestro programa por consola.



EJERCICIO 75

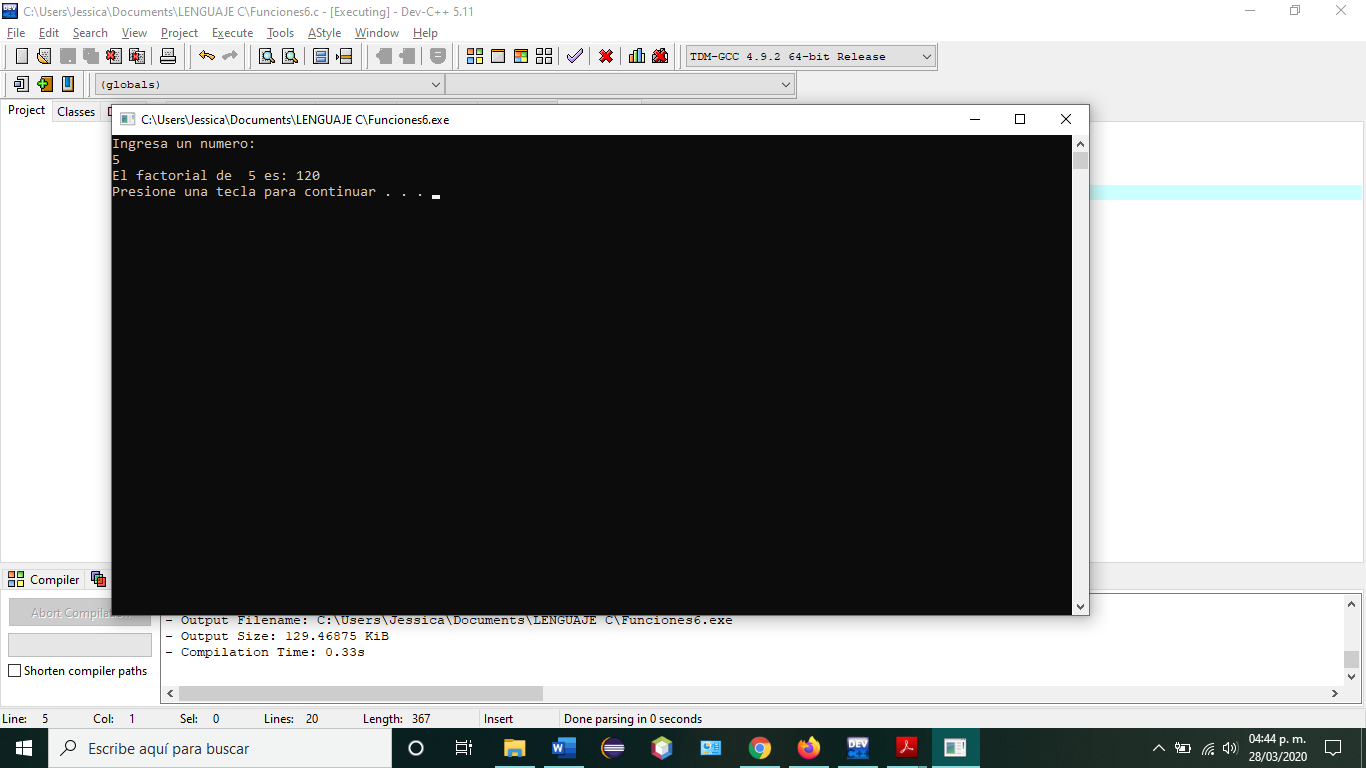
Haz un programa con funciones recursivas que calcule el factorial de un número n ingresado desde teclado.



Damos clic en el botón de compilar y ejecutar, o usando la función F11.



Se nos muestra nuestro programa por consola.



**V. Conclusiones:**

Aprendí más cosas sobre Lenguaje C, el uso de cadenas y funciones, su declaración y su implementación.

Entendí el concepto de cadena y una función en C.