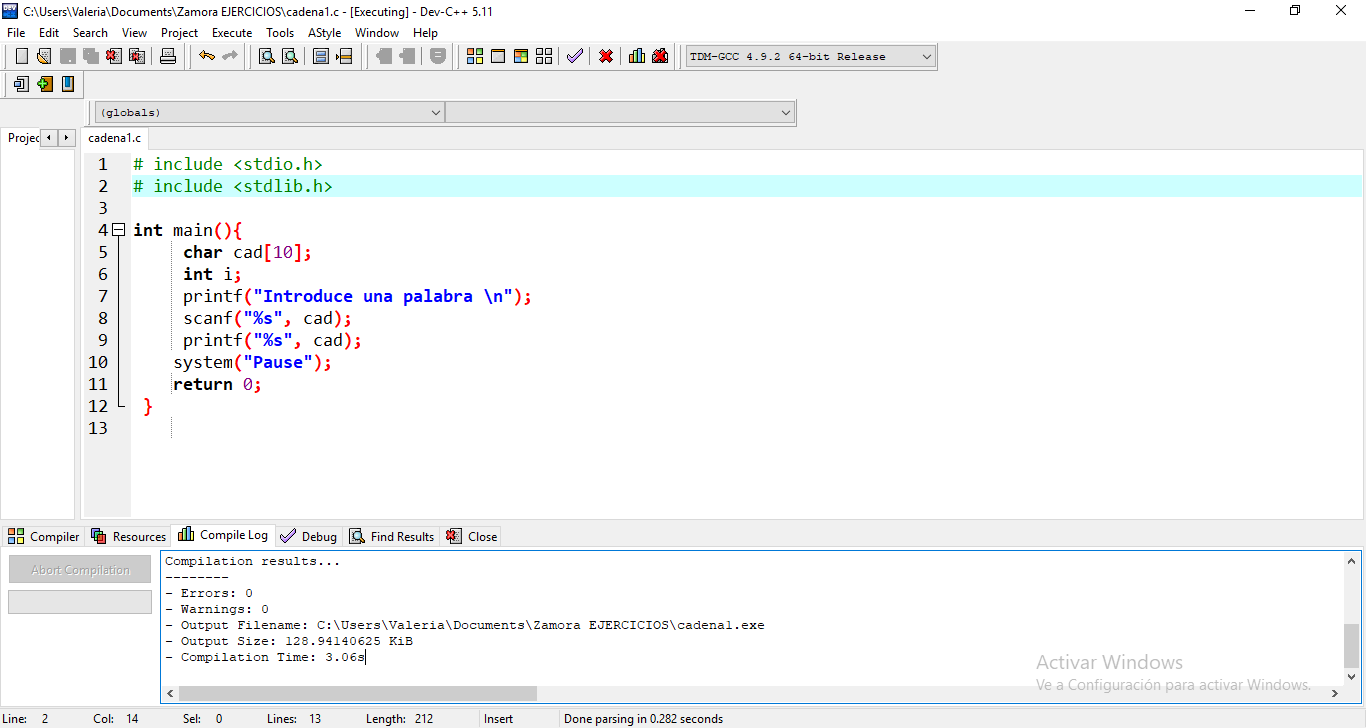
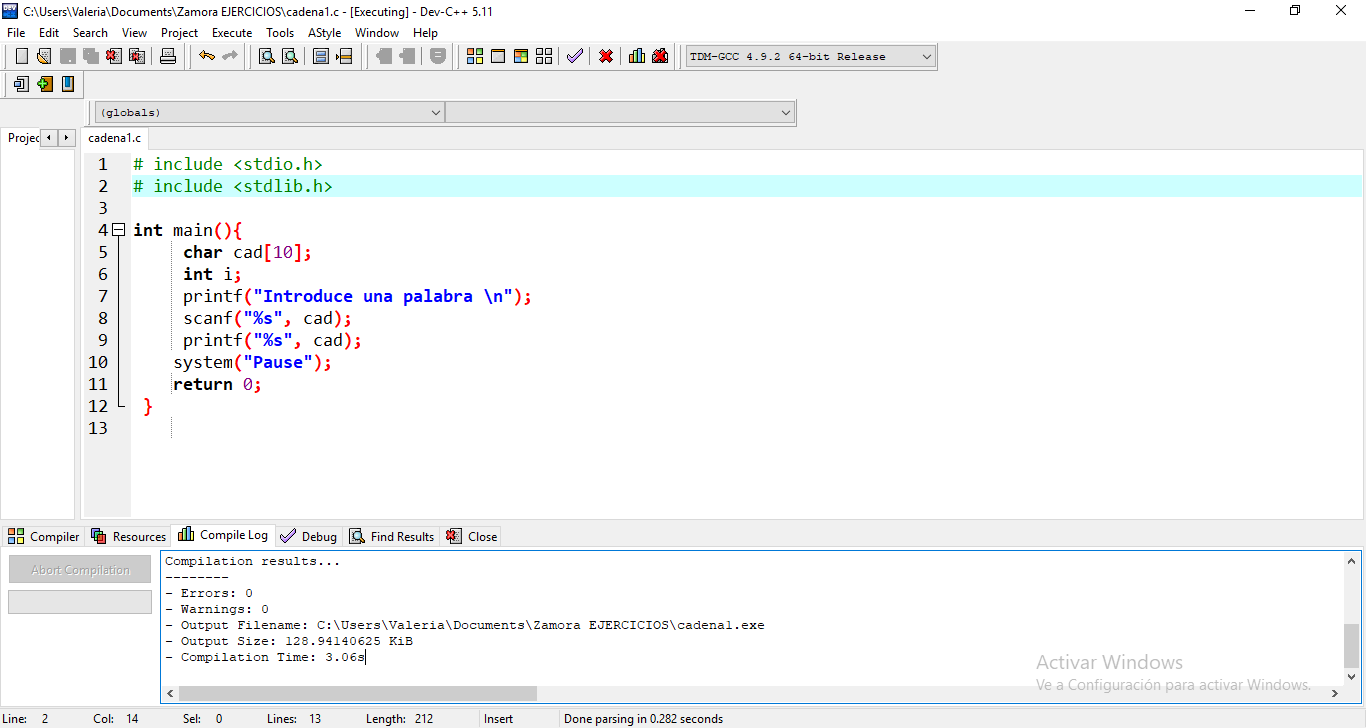
**CADENAS**

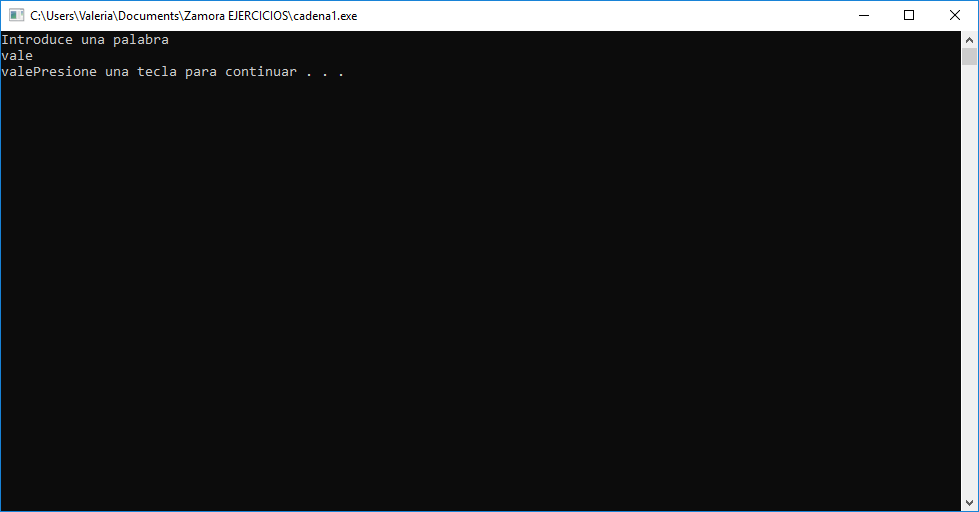
**¿Qué es una cadena?** Es un arreglo de caracteres y por lo general empieza “\0”.





Para ejecutar clic en el botón de compilar y ejecutar

Por consola nos da como resultado:

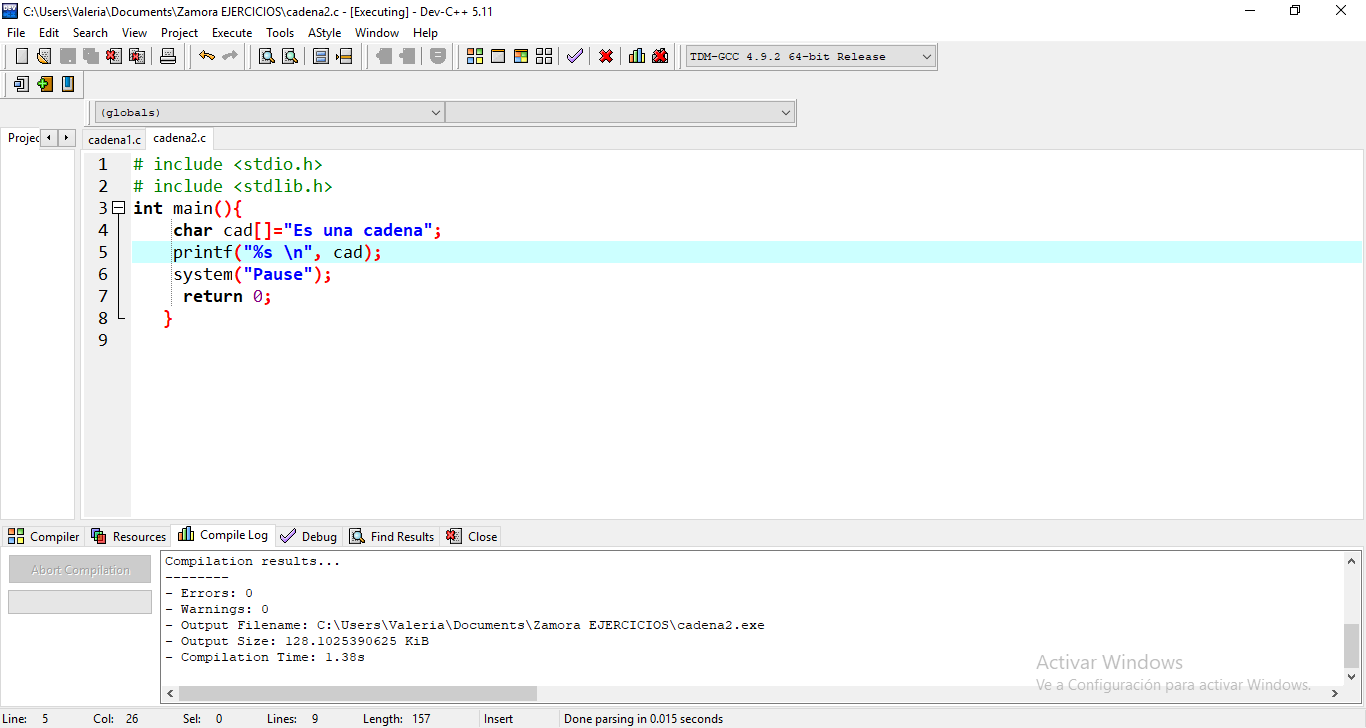


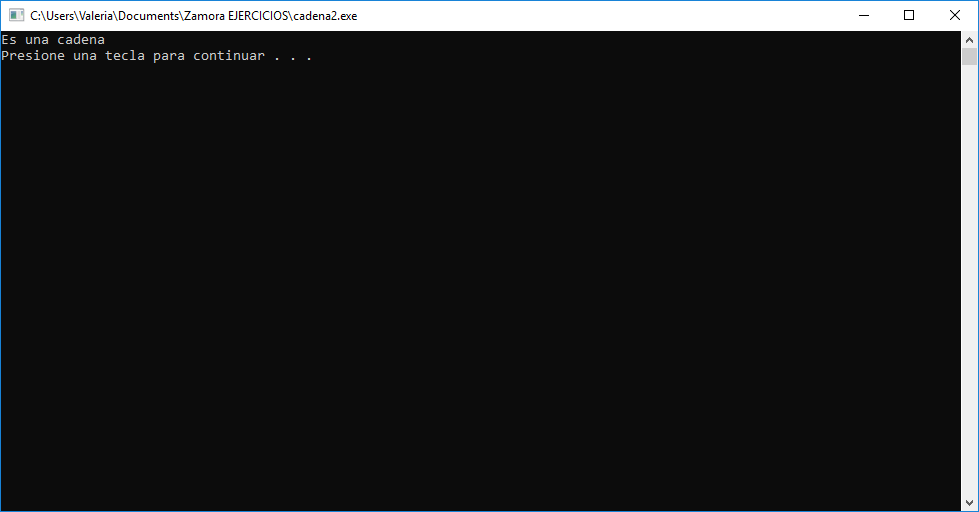
Un arreglo sin tamaño:

** char cad [] = “Es una cadena”;**

Los elementos consta el arreglo cad son de 14.

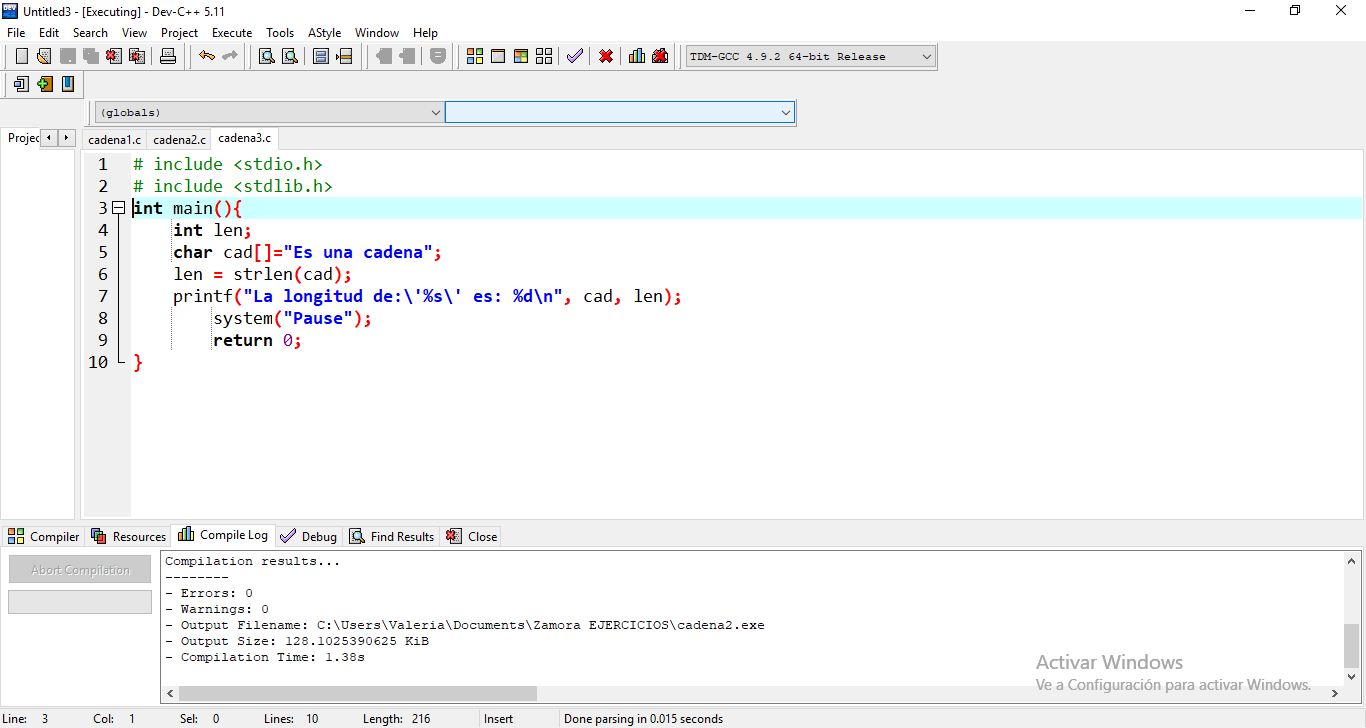
Ejercicio:

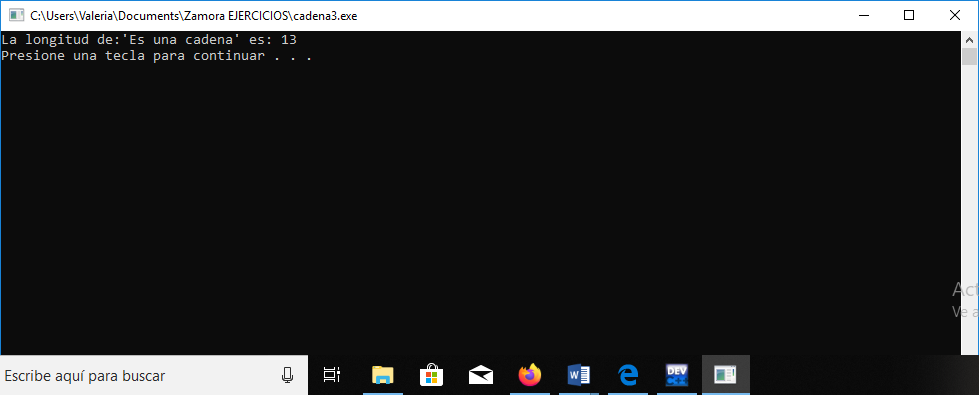




Devuelve la longitud de la cadena sin tomar en cuenta el carácter de final de cadena.

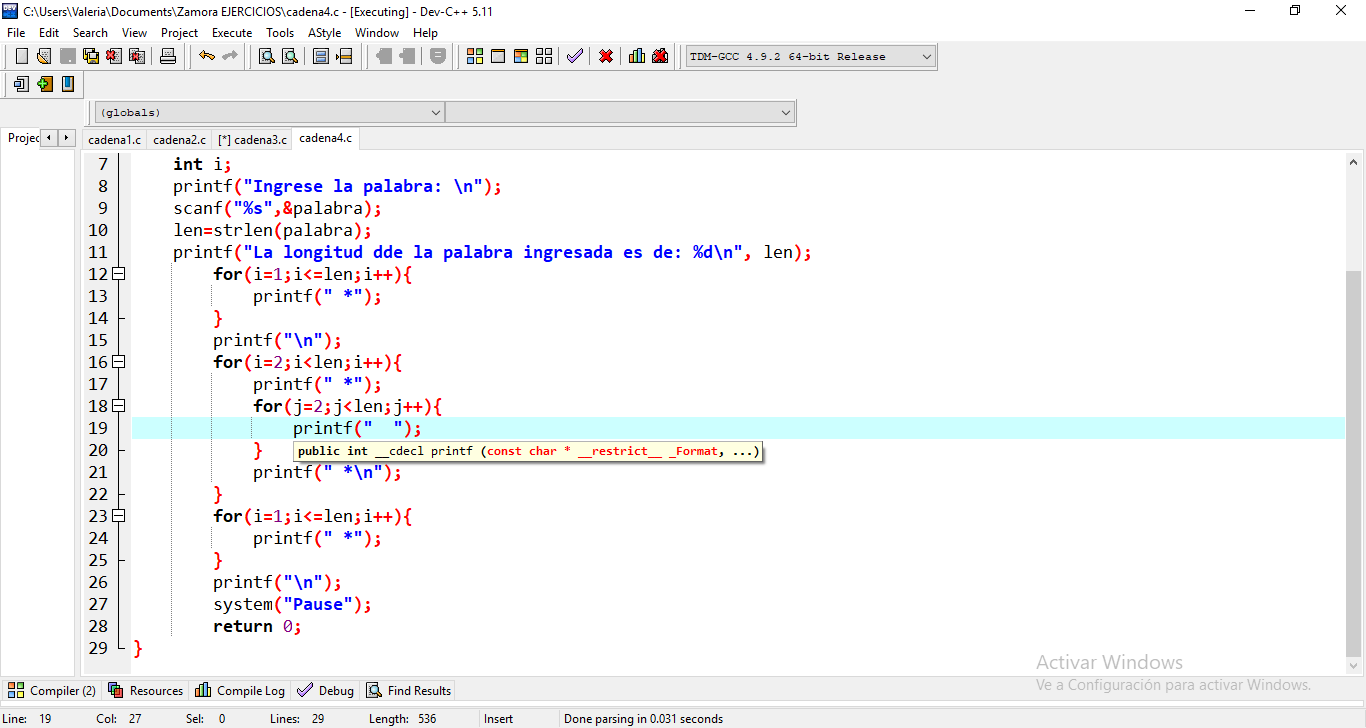
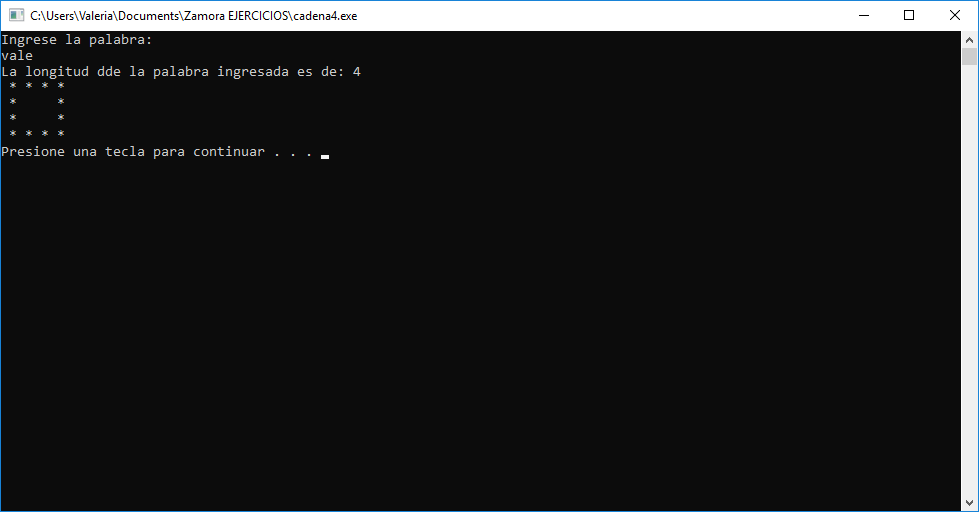
**strlen(<cadena>)**

Ejemplo:

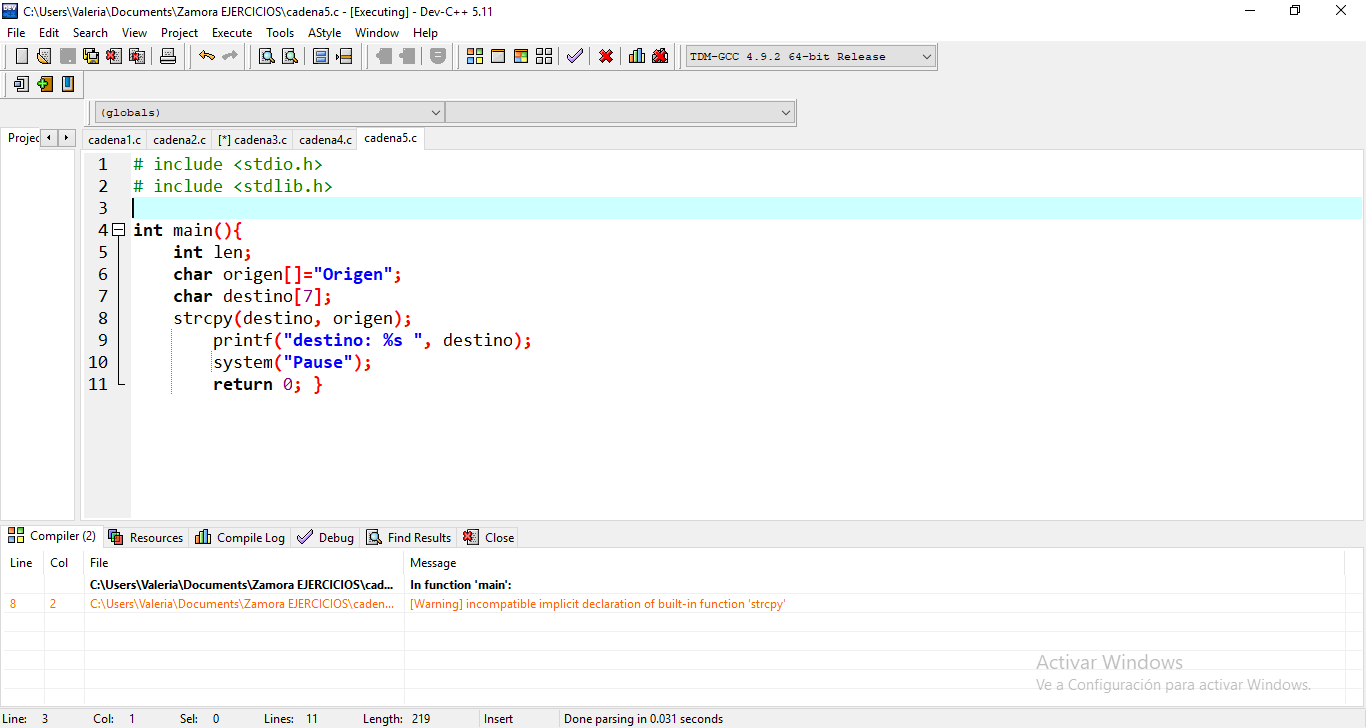
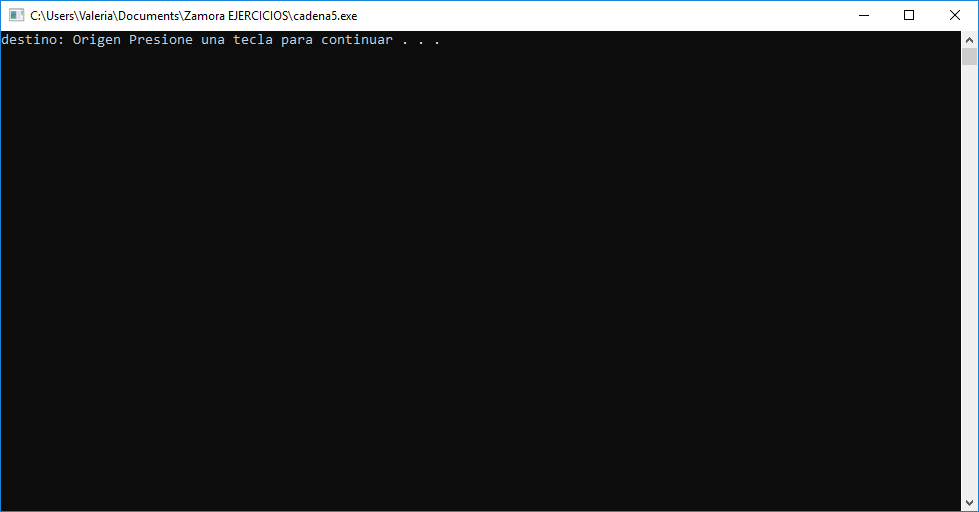


Ejercicio:

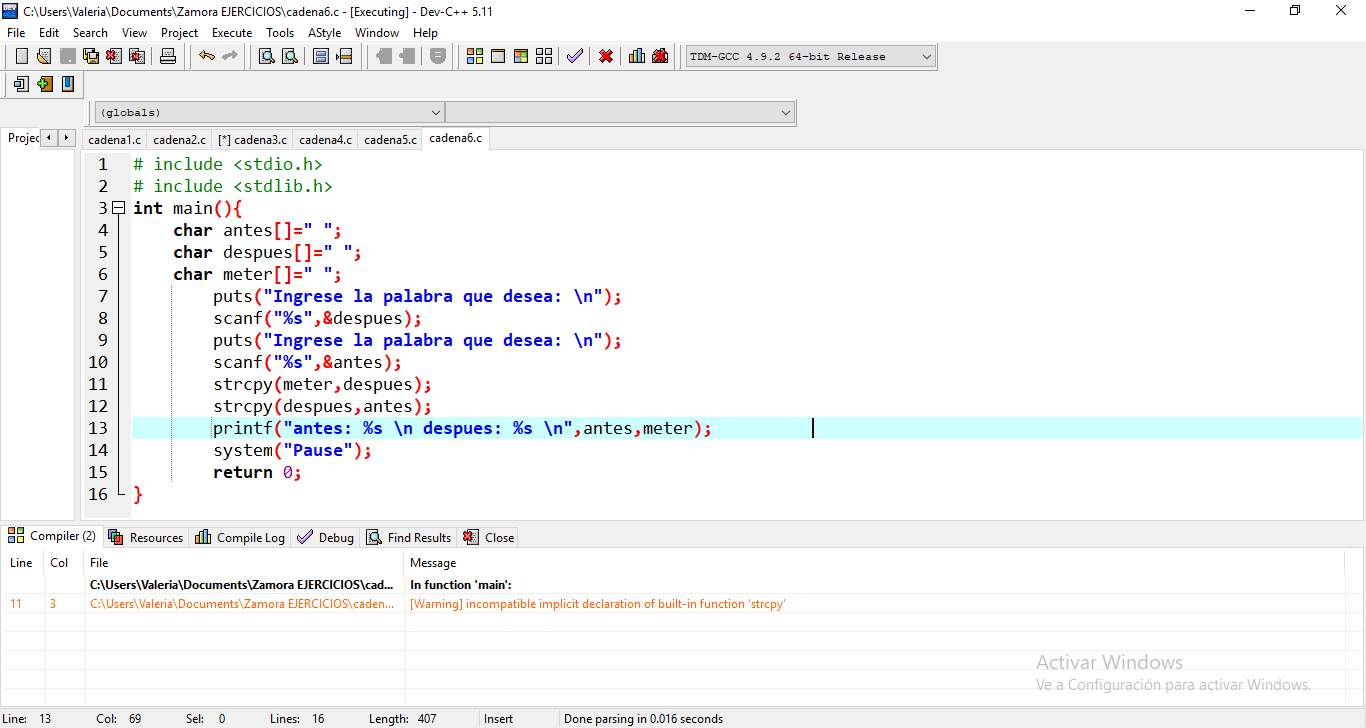
Escribe un programa que reciba una palabra por teclado. De acuerdo a la longitud de la palabra (N) que se ingresó por teclado imprime un cuadrado de asteriscos de (N x N).

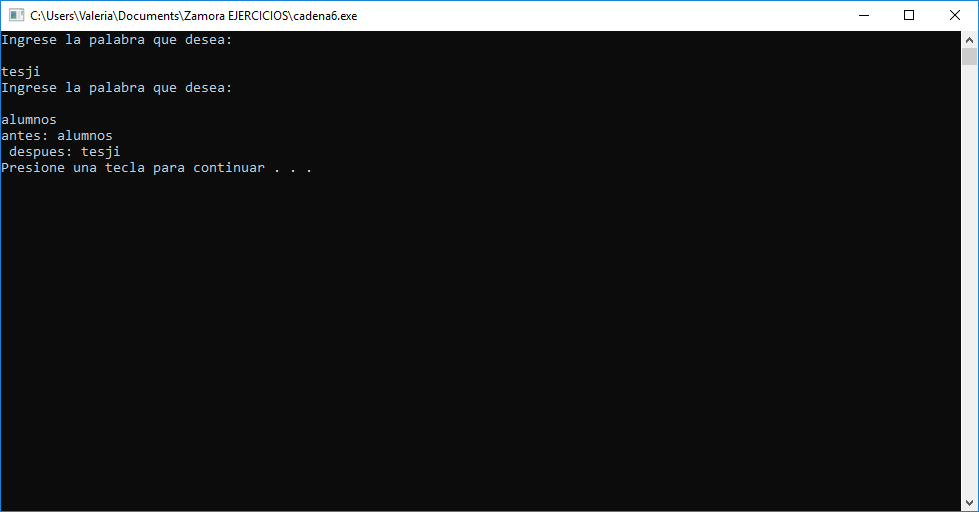


Ejemplo:

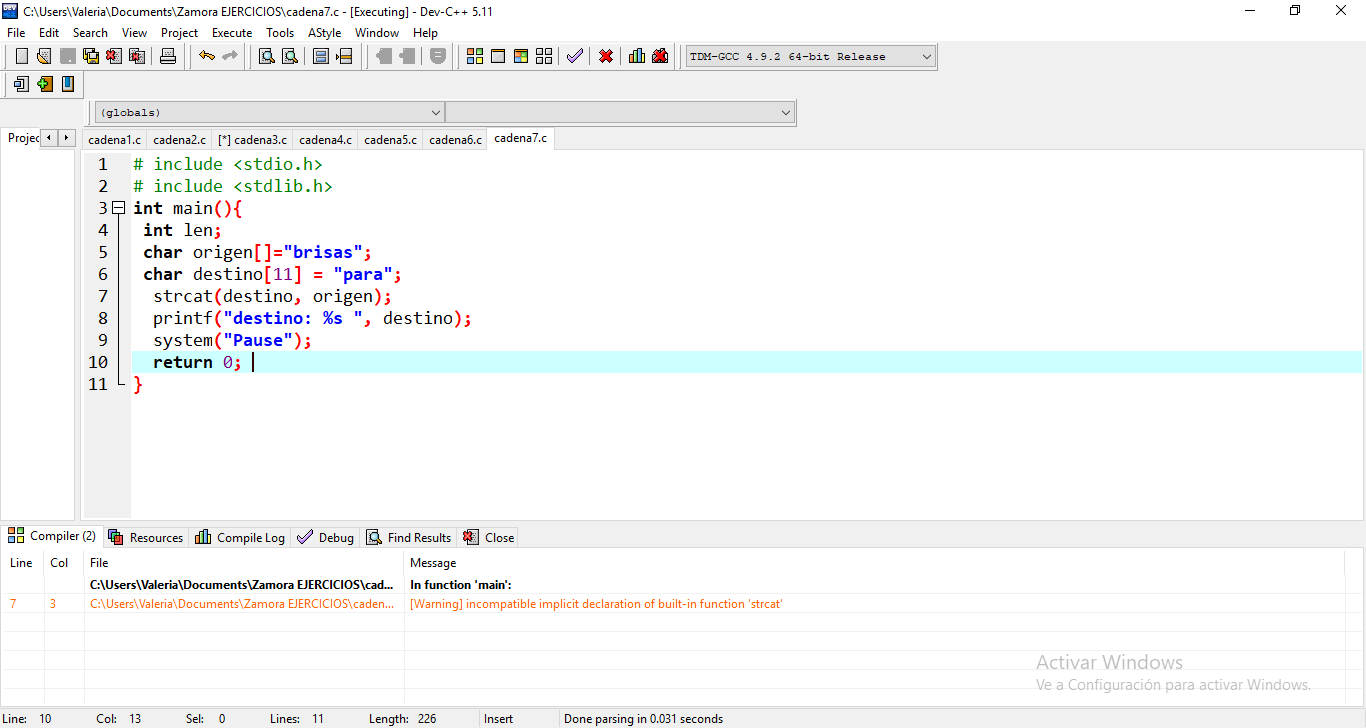


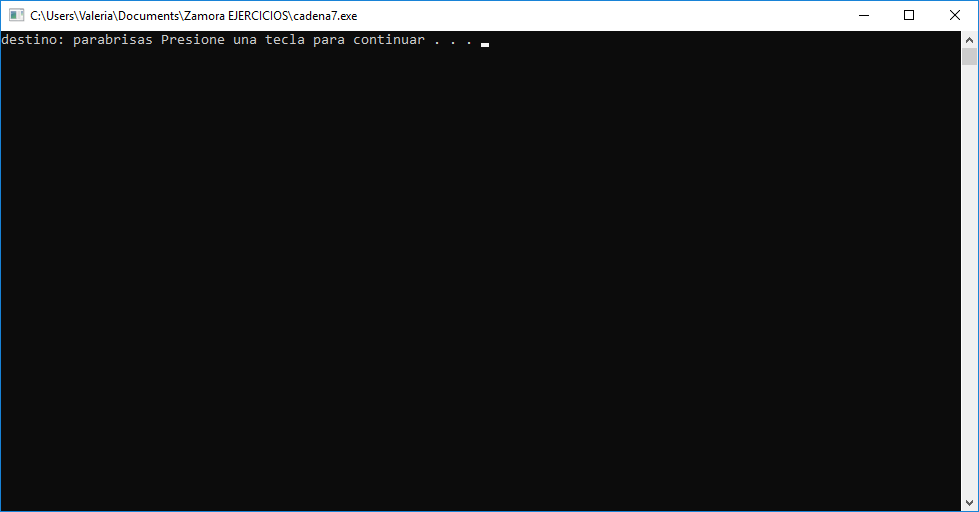
Ejercicio: Escribe un programa que reciba por teclado dos palabra y cada una de ellas las almacene en un arreglo. Después intercambia sus contenidos. Imprime el antes y el después



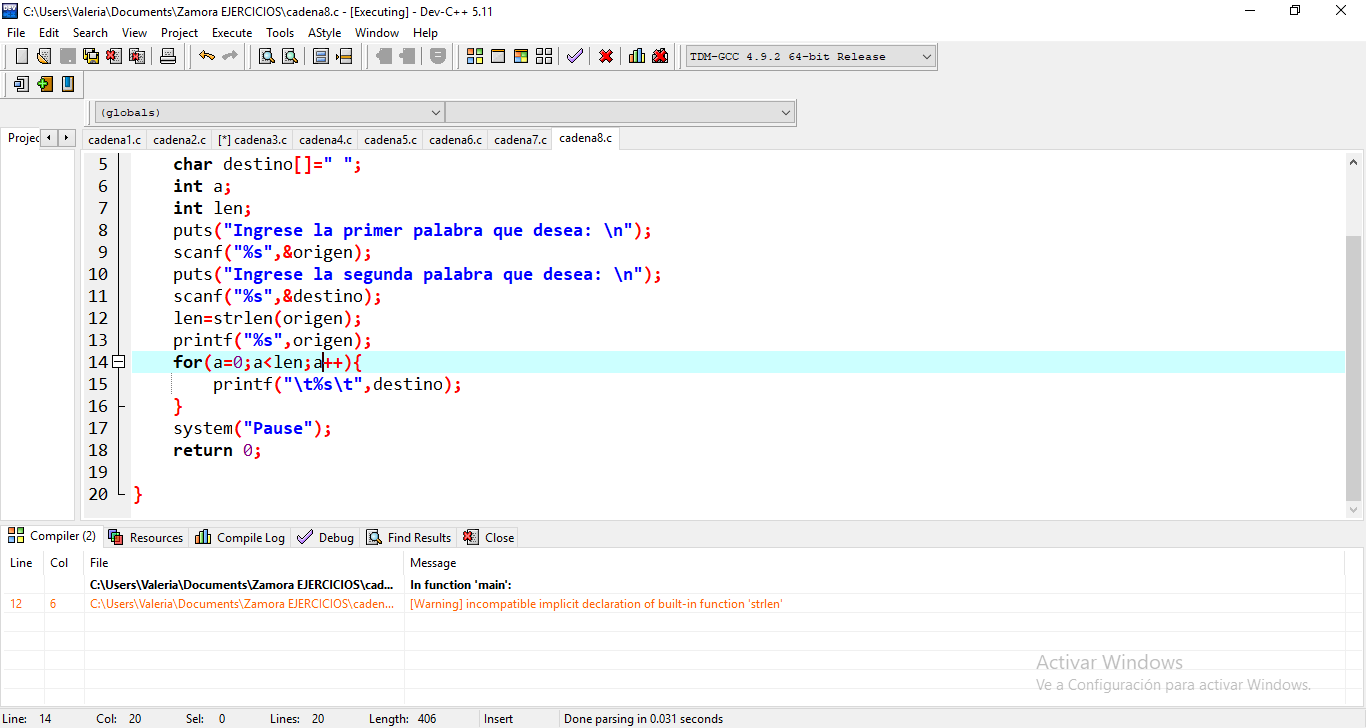


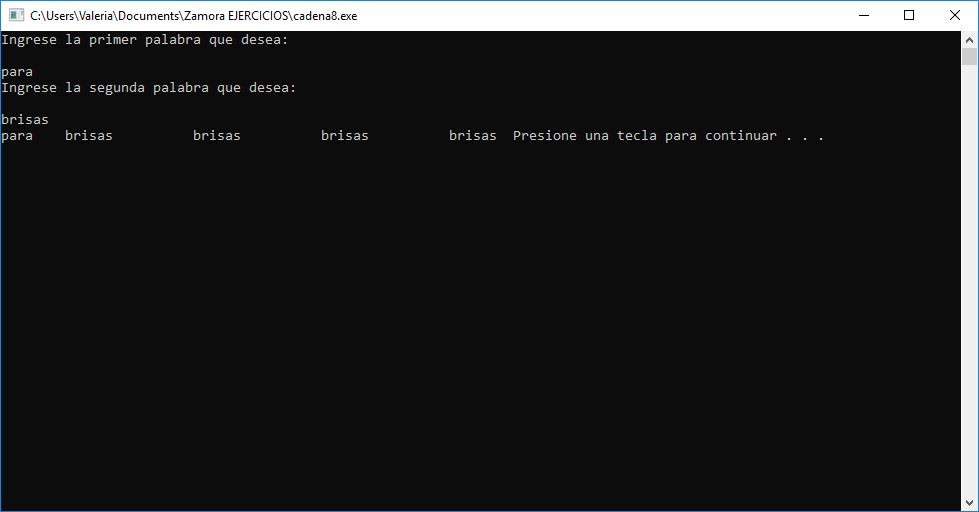
**Funciones de cadena: strcat**

Ejemplo:

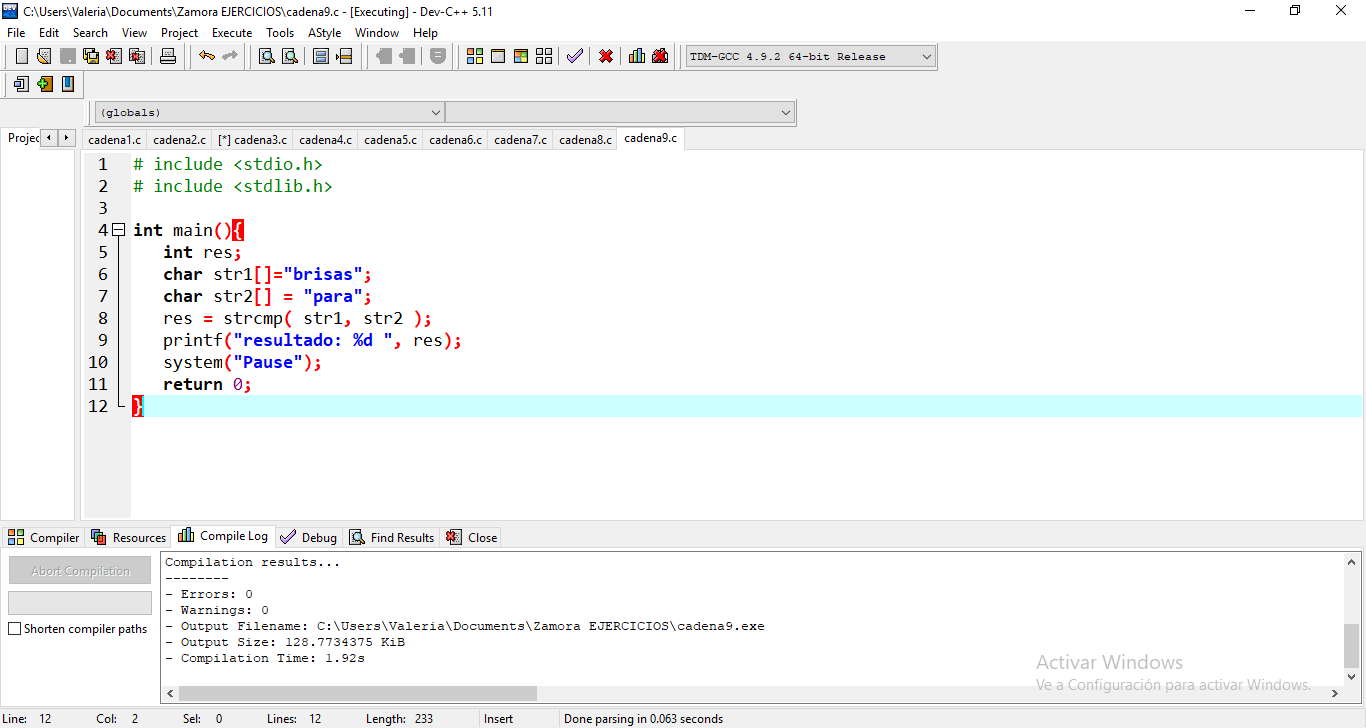


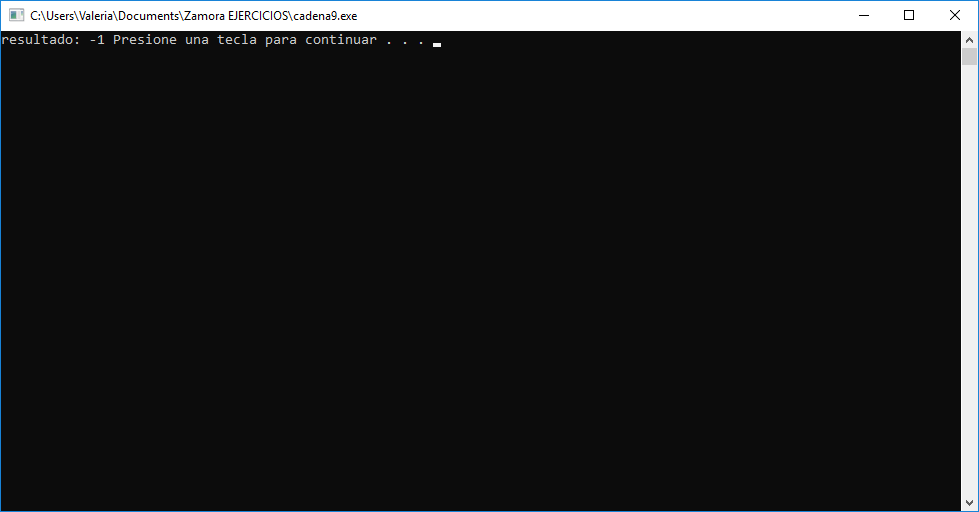
Ejemplo: Escribe un programa que reciba por teclado dos palabras.  Y concatene N veces la segunda palabra a la primer palabra. Donde N es la longitud de la primera palabra.

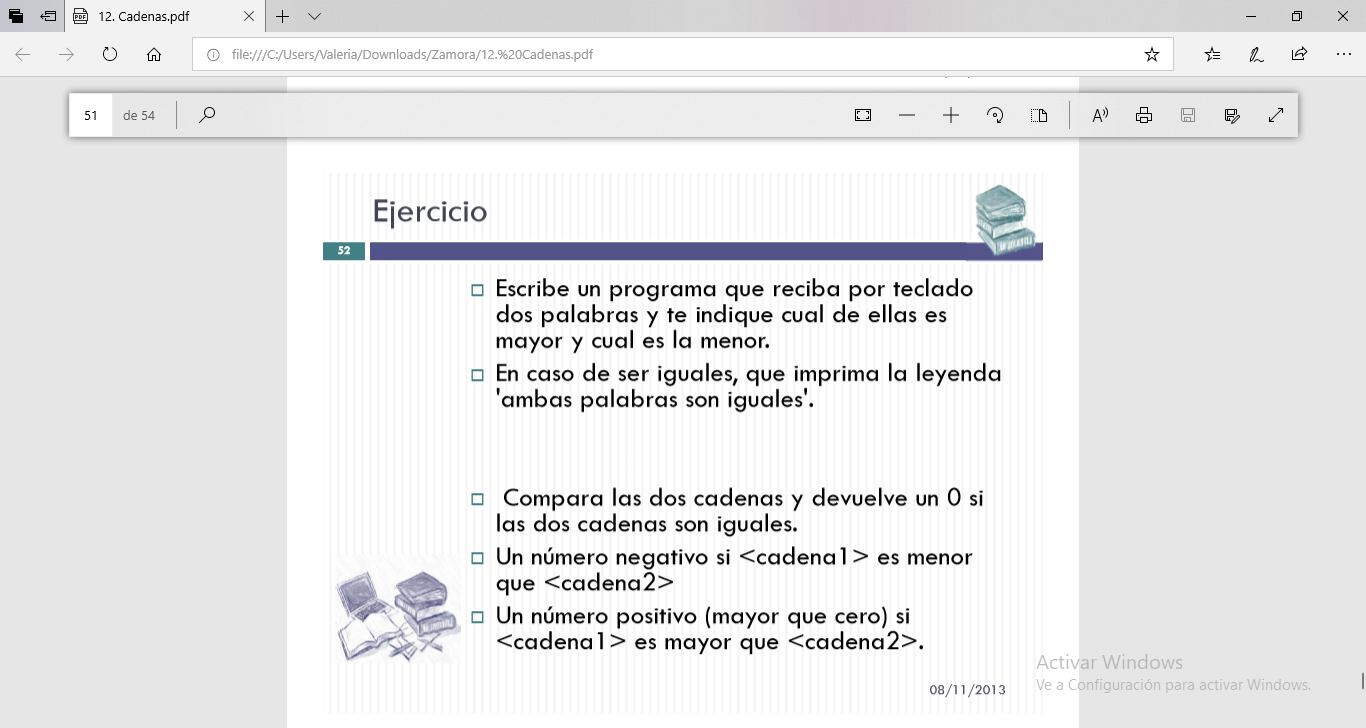


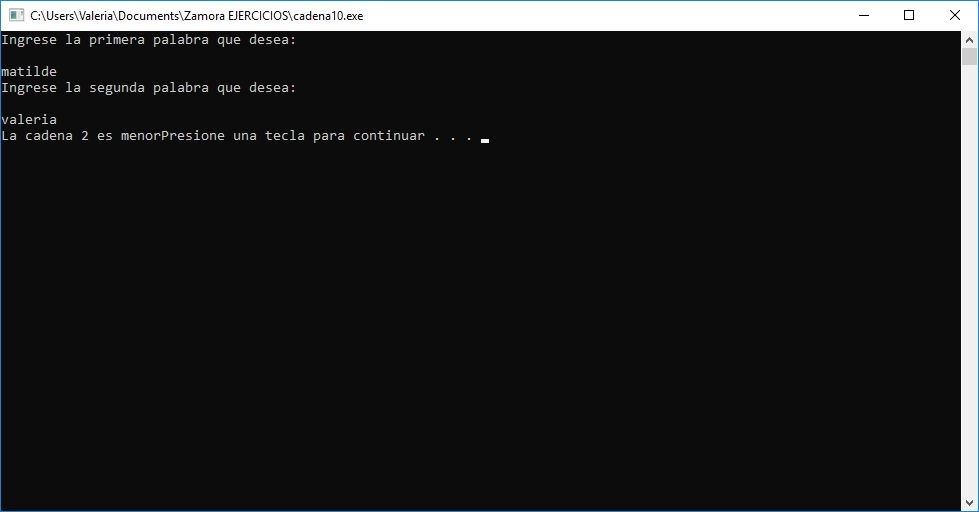
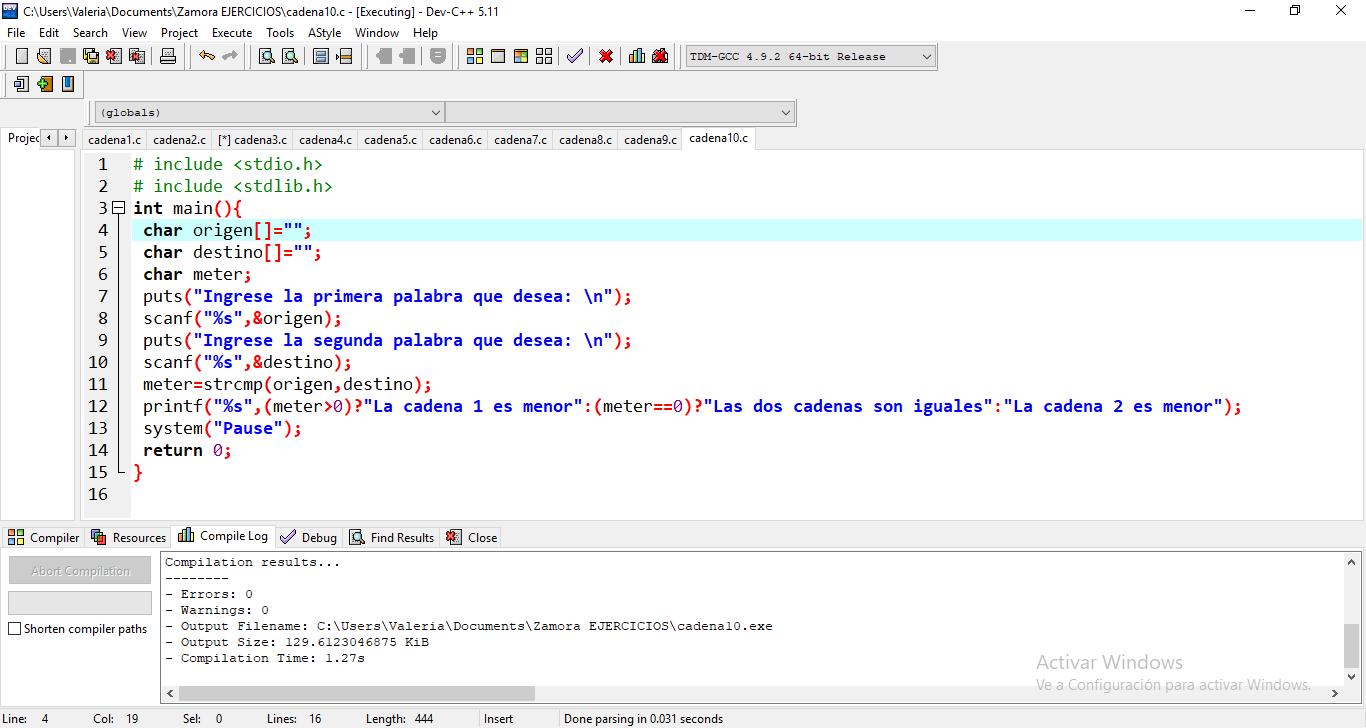


**Funciones de cadena: strcmp**

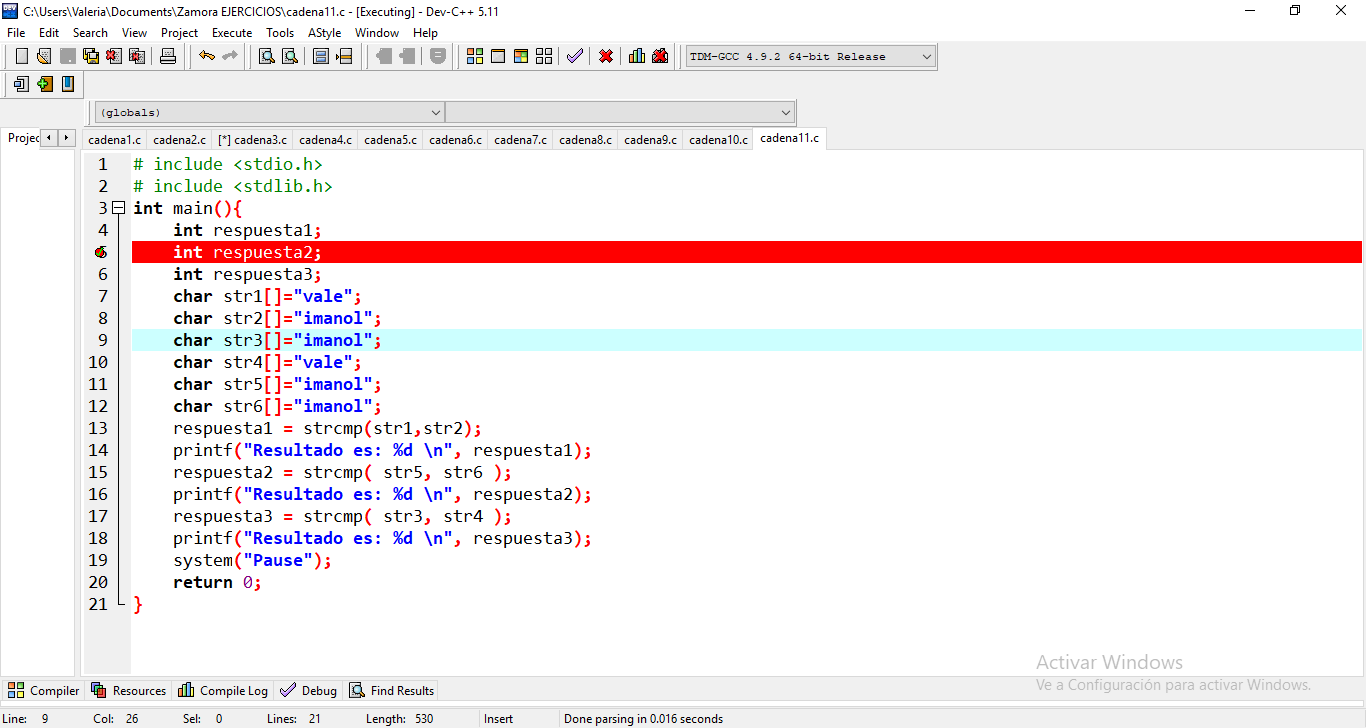
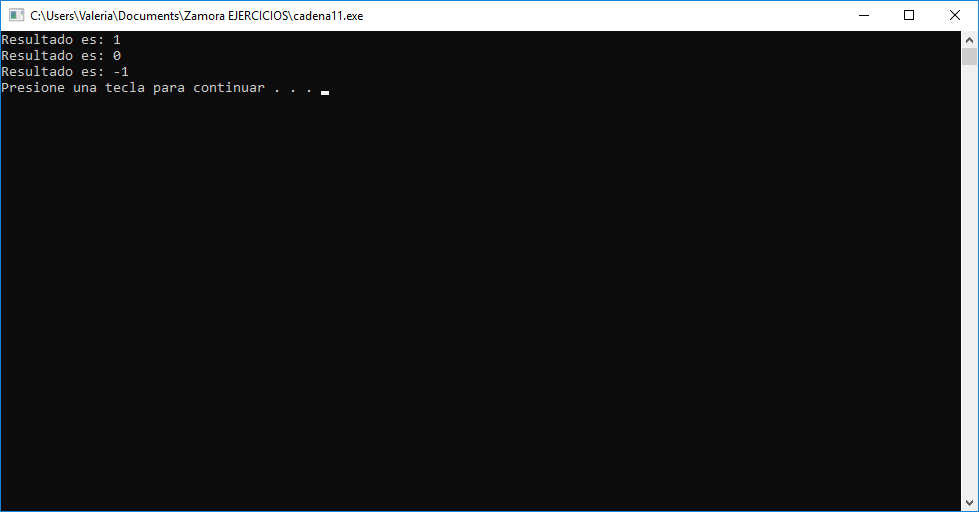
Compara las dos cadenas y devuelve un 0 si las dos cadenas son iguales.





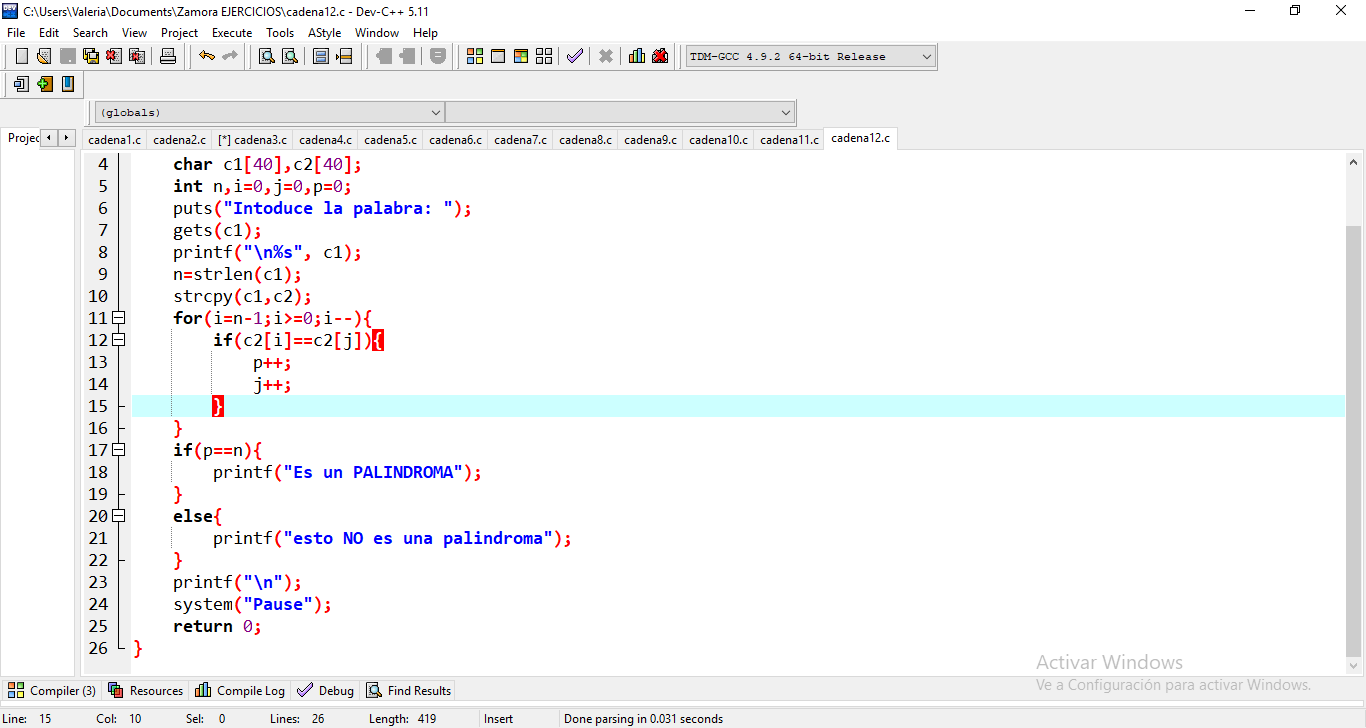


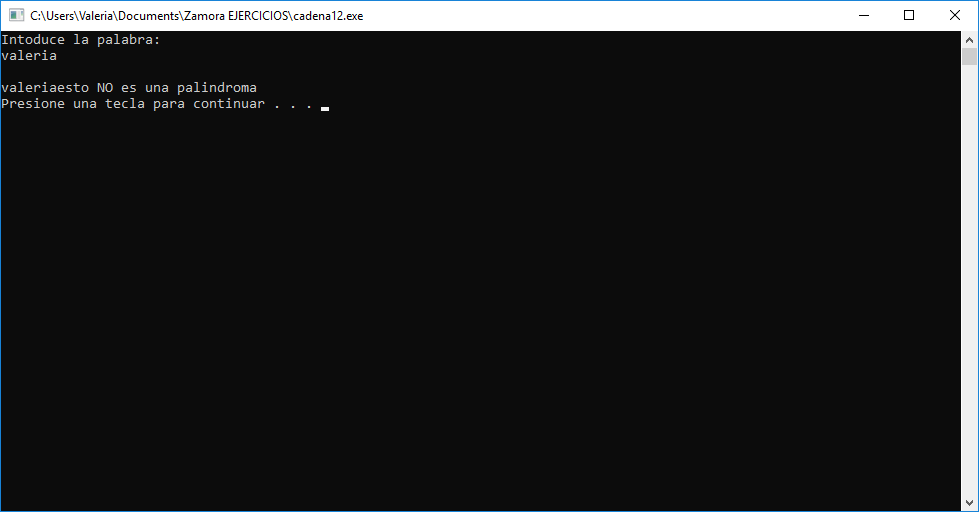
Ejercicio: Compara las dos cadenas y devuelve un 0 si las dos cadenas son iguales. Un número negativo si <cadena1> es menor que <cadena2>  Un número positivo (mayor que cero) si <cadena1> es mayor que <cadena2>.



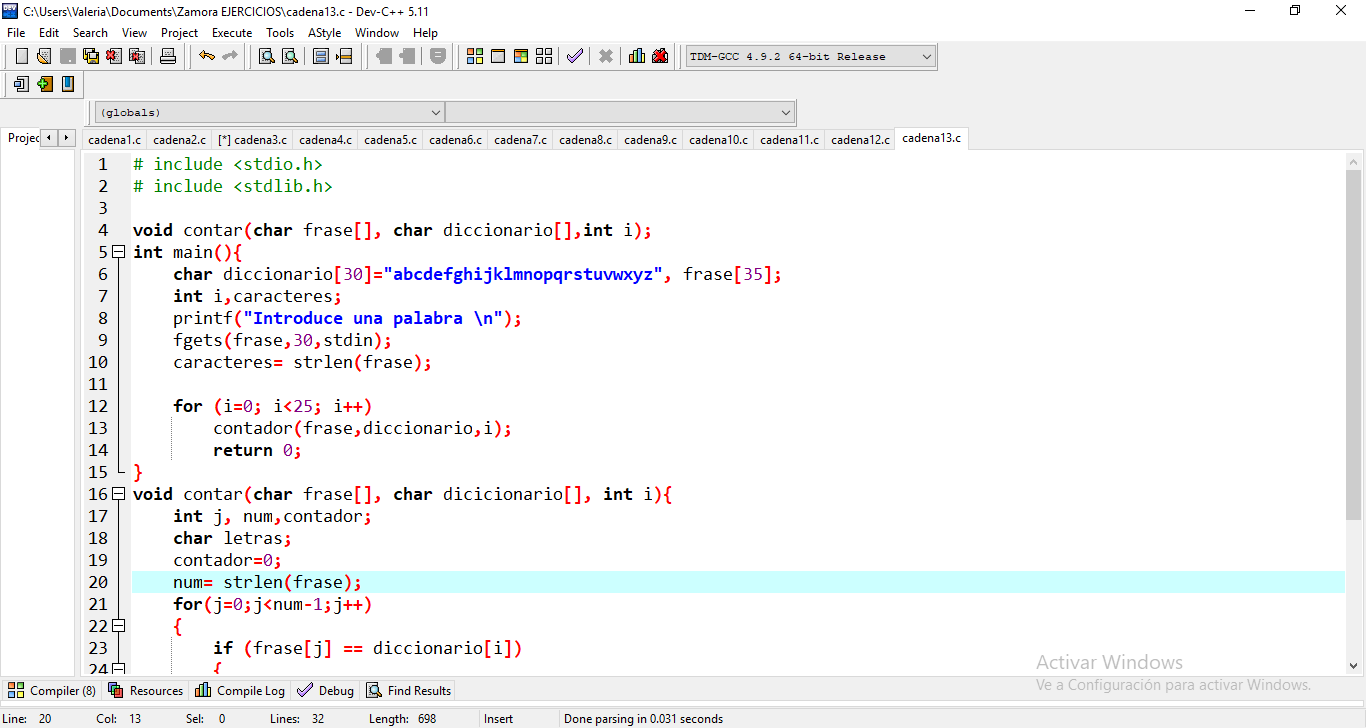
Ejercicio: Crea un programa que detecte una palabra palíndroma. Los palíndromos son palabras que se leen igual de izquierda a derecha que de derecha a izquierda.

Ejemplo: ala, rotor, salas





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

}

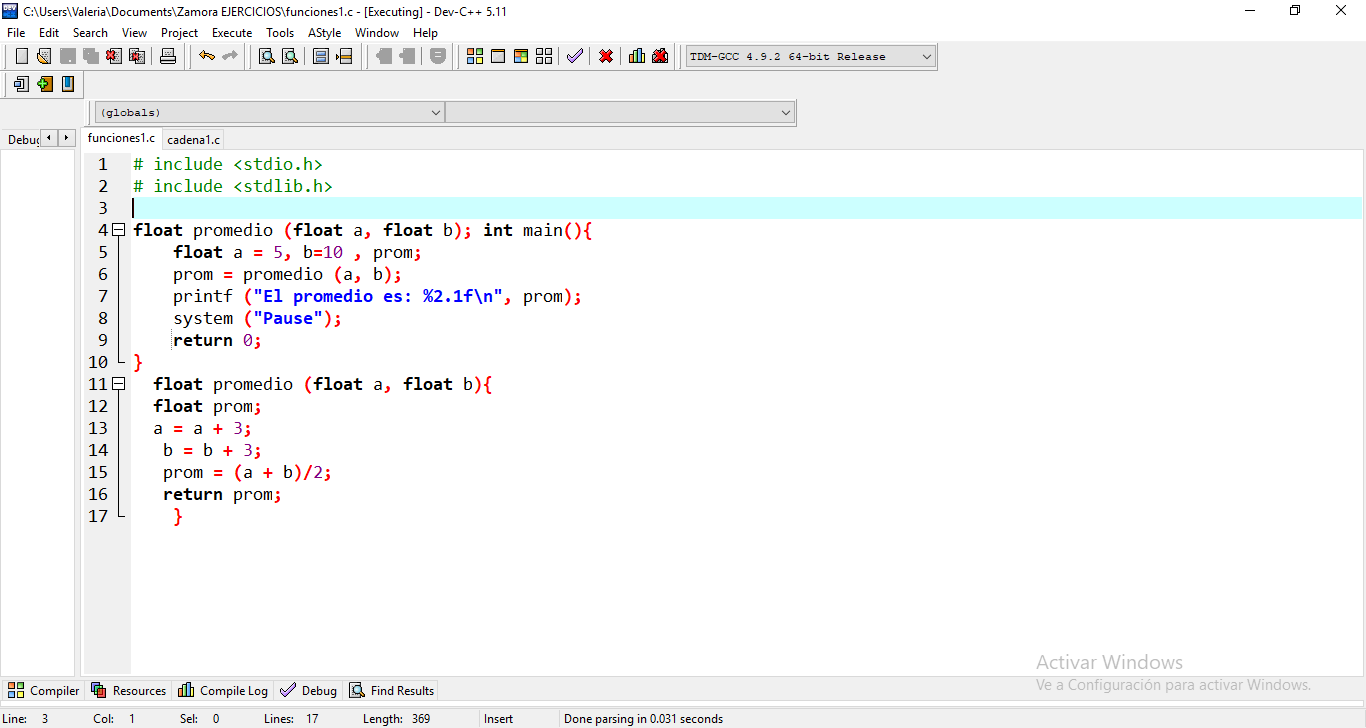
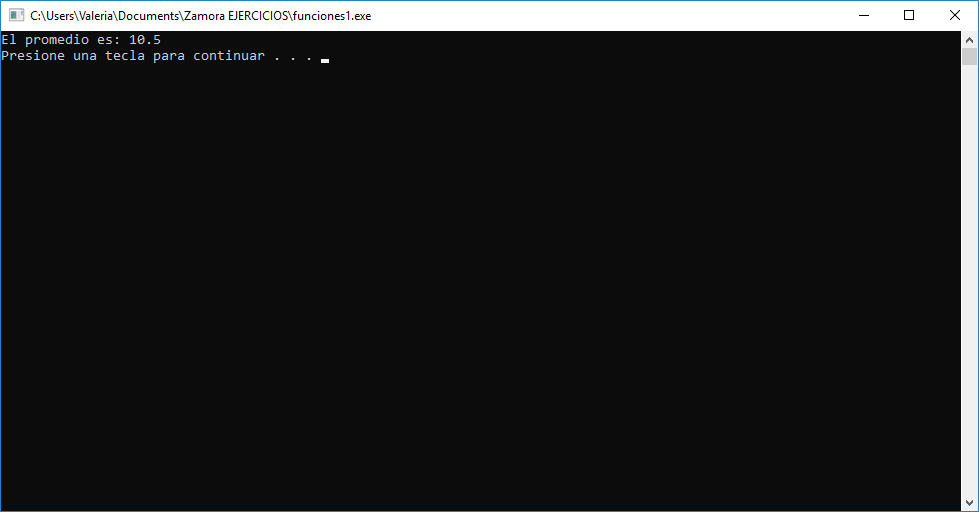
**FUNCIONES**

Definición de función matemática: Es una operación que toma uno o más valores llamados argumentos y produce un valor llamado resultado.

Para la **declaración** de funciones es:

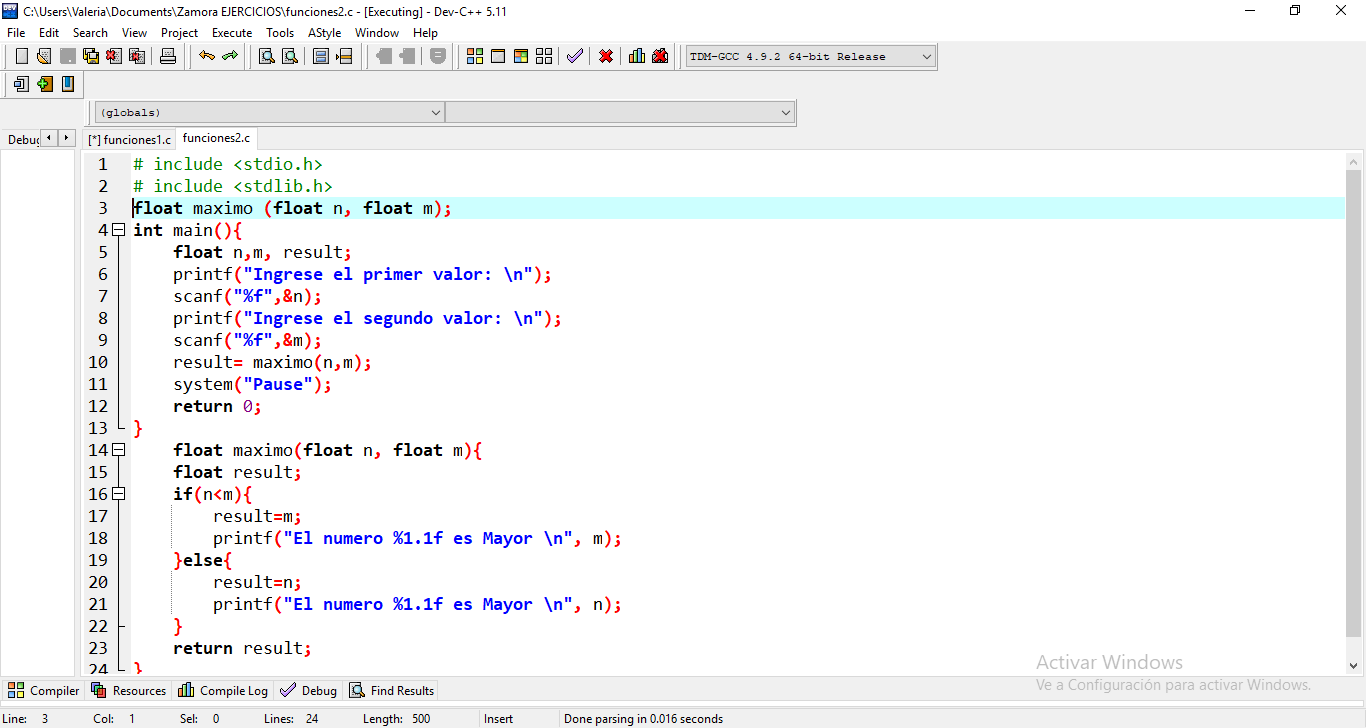
**tipo : especifica el tipo de valor que devuelve la función. Si no se especifica tipo, el compilador asume que es entero (int).**

Ejercicio:

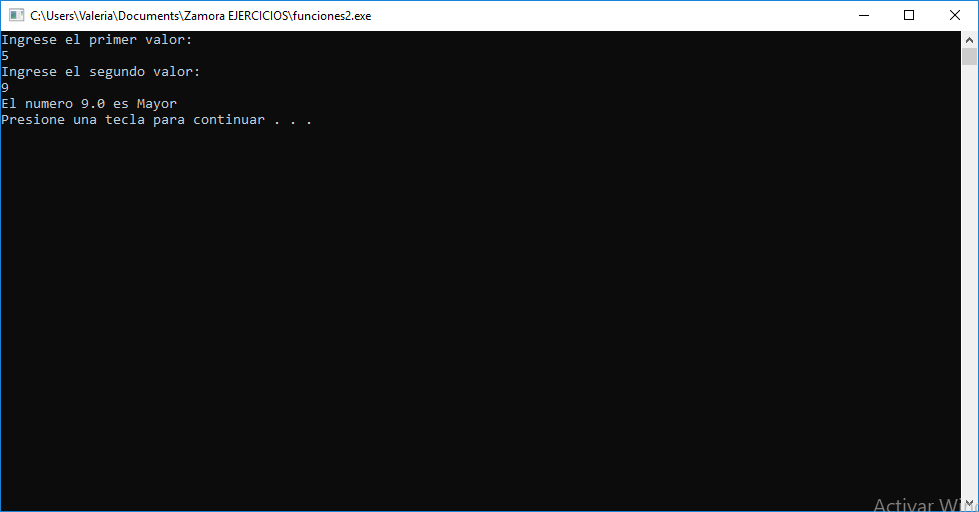


**Variables locales y globales**

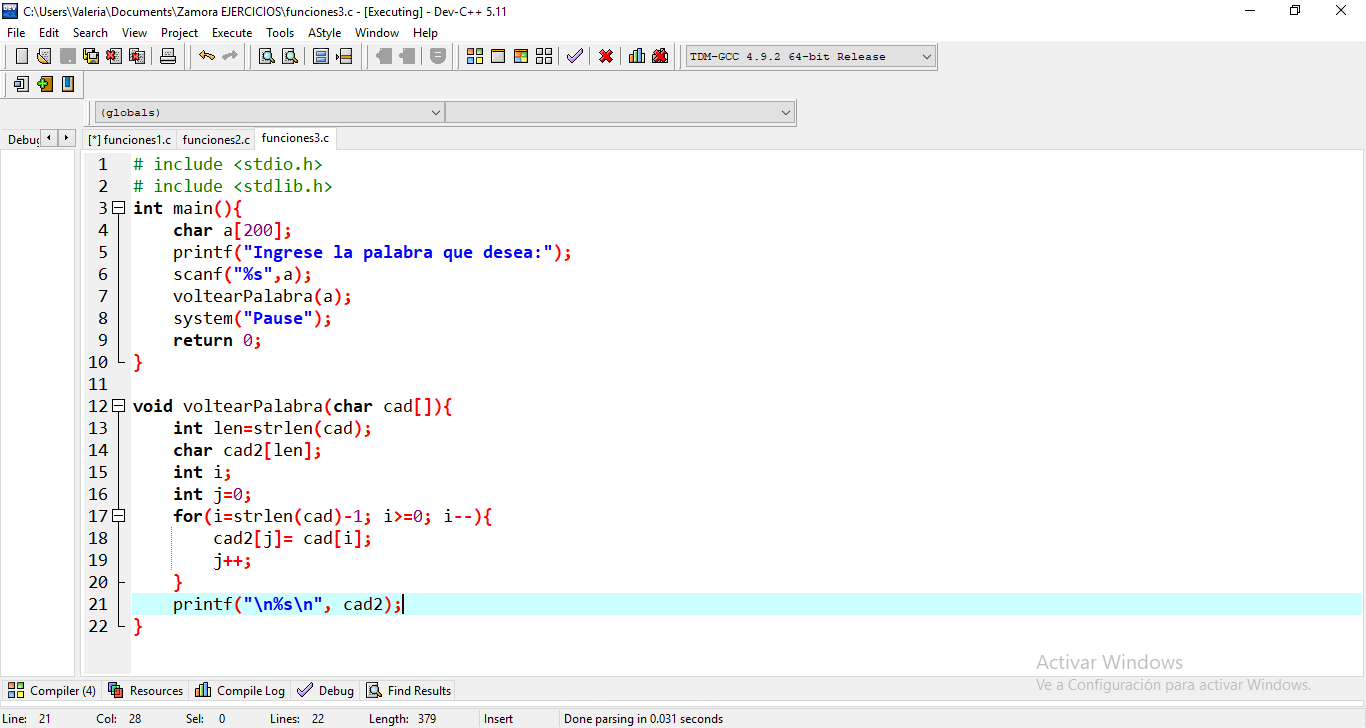
Variables Locales: Se declaran dentro de la función y sólo están disponibles durante su ejecución.

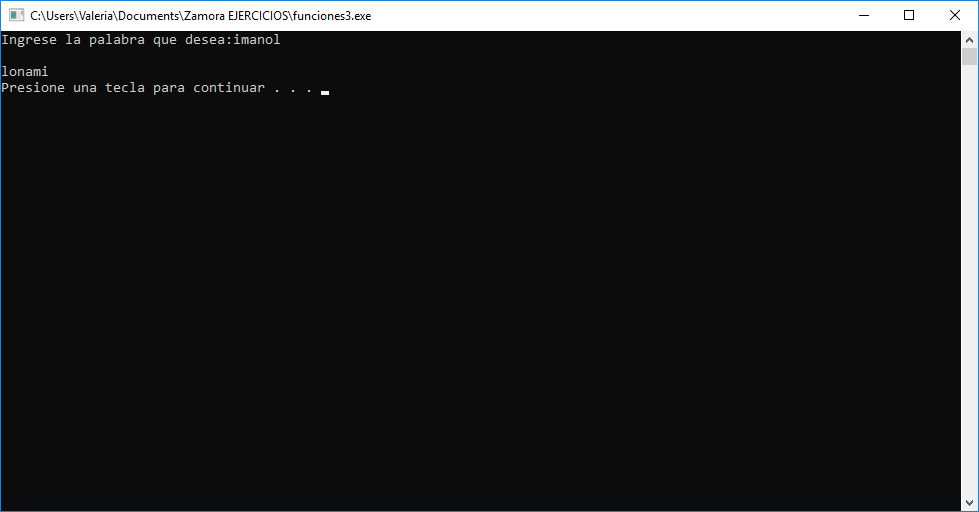
 Variables globales: Se declaran fuera de las funciones. Pueden ser utilizadas por todas las funciones.

Ejercicio:

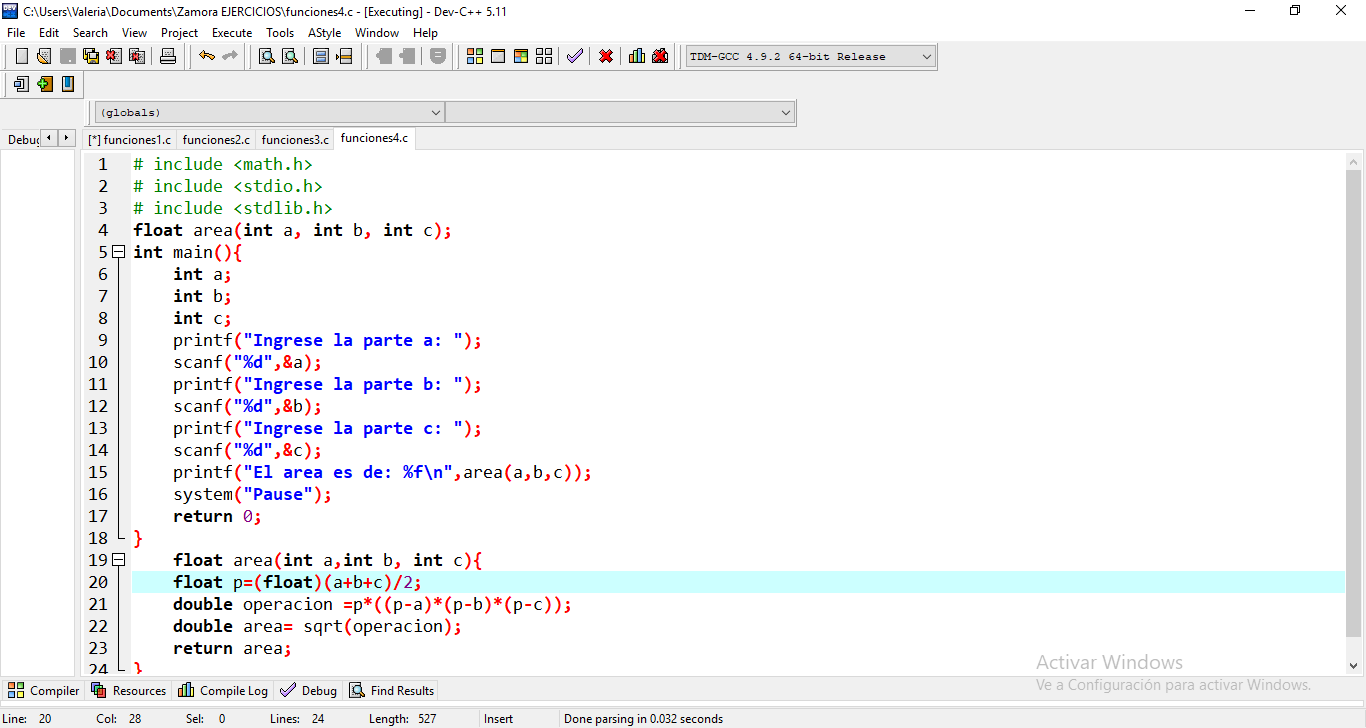


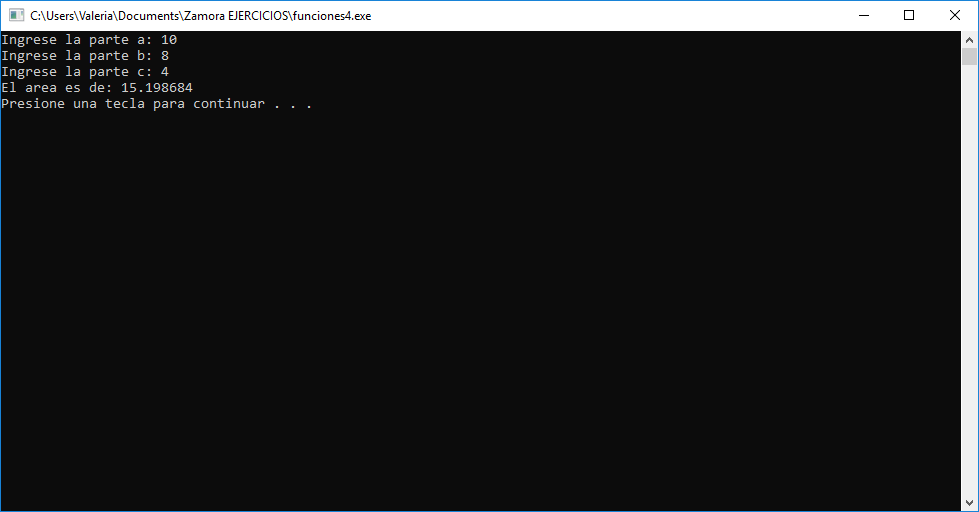
Ejercicio:





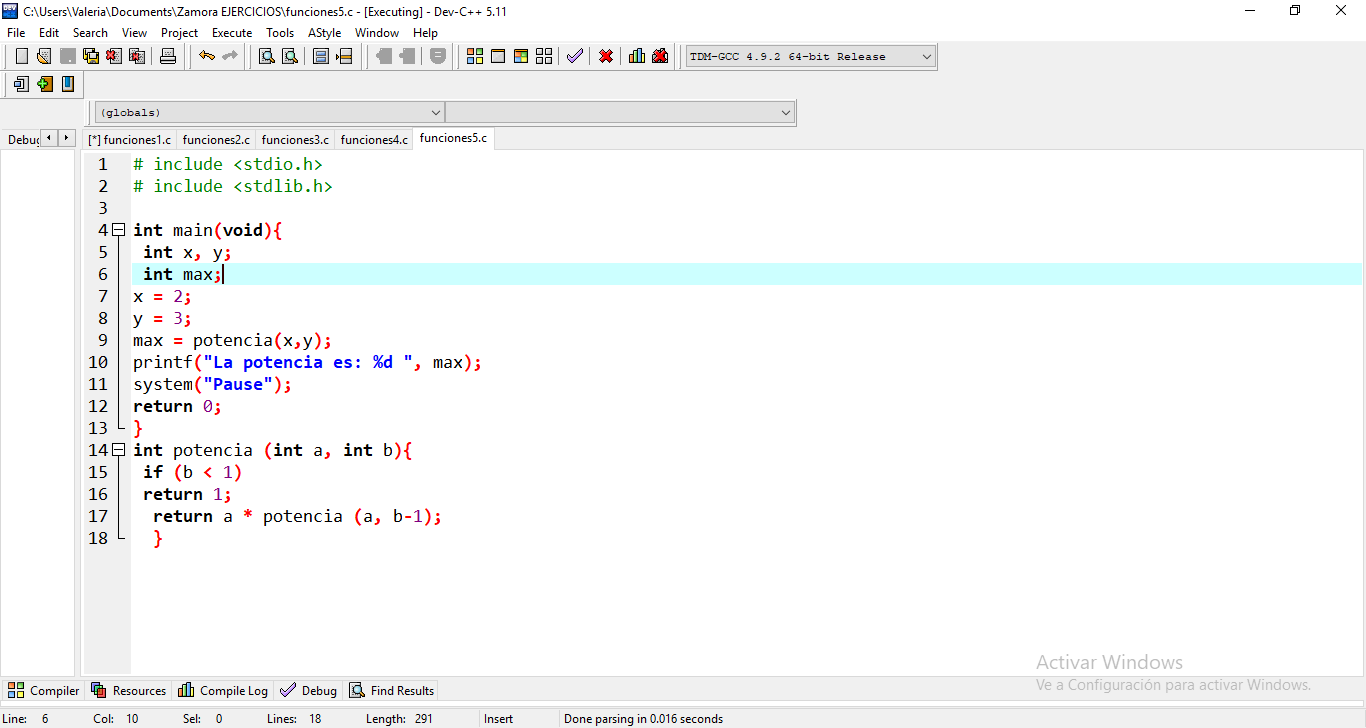
Ejercicio:

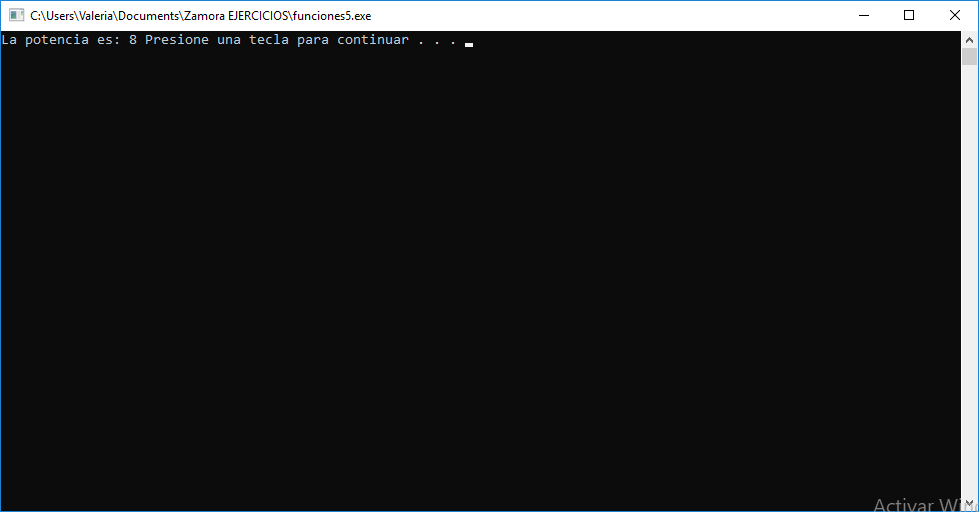




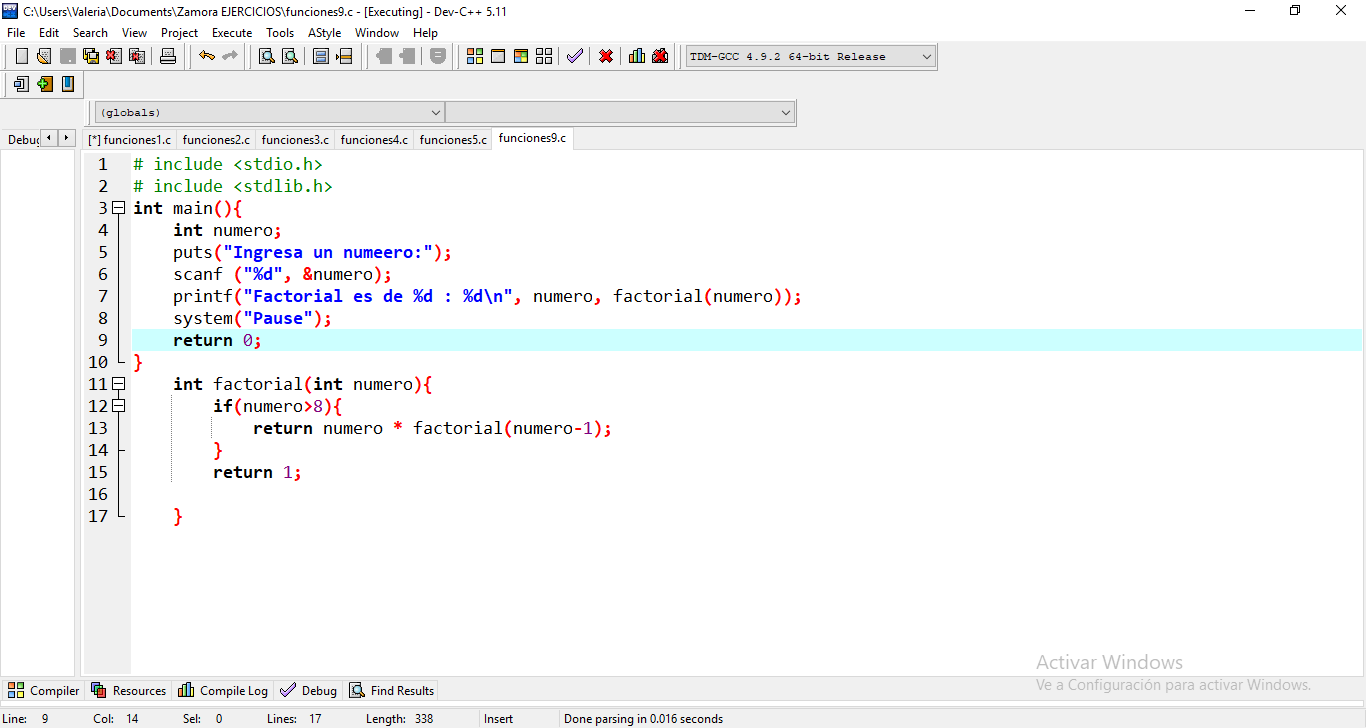
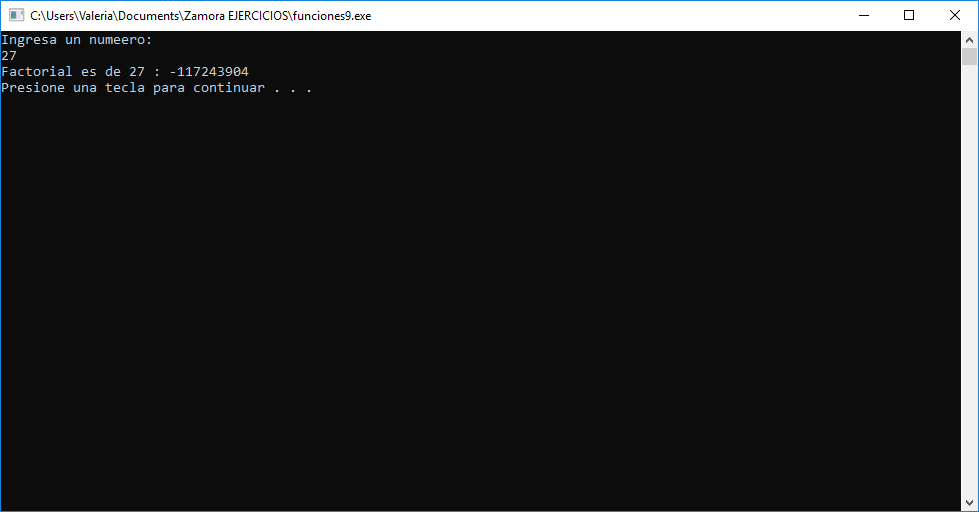
**Funciones recursivas**

Son aquellas que se llaman a sus mismas de forma repetida hasta que se cumpla alguna condición.





Ejercicio:



Conclusiones: Trabajar en C++ es interesante porque hay varias maneras de resolver los problemas y la práctica ayuda mucho para tener la capacidad de poder resolver con mas dificultad.