|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre de la práctica** | **Cadenas y Funciones** | | | **No.** |  |
| **Asignatura:** | **Métodos Numéricos** | **Carrera:** | **ISIC** | **Duración de la práctica (Hrs)** |  |

**Emiliano Arce de Jesús Gpo.3402**

1. **Competencias especificas**

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

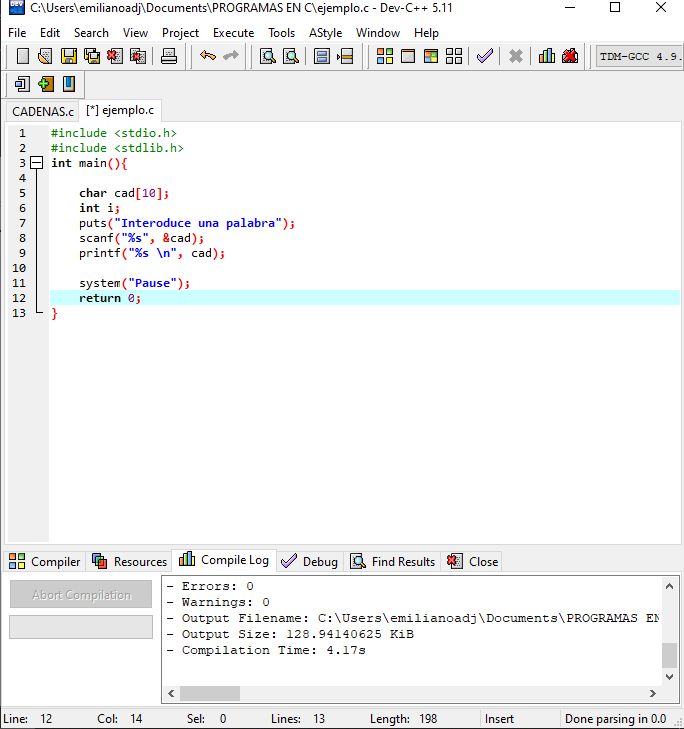
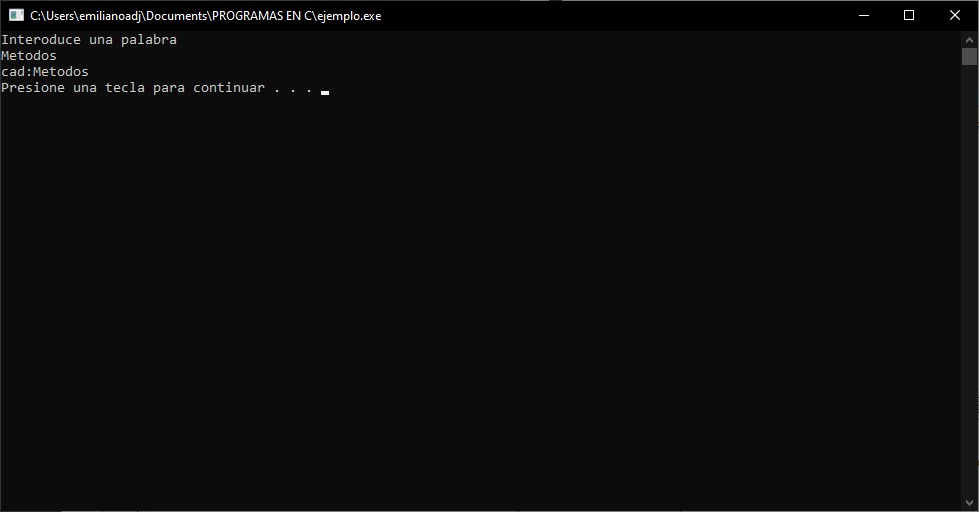
1. **Material empleado:**

* **Lenguaje de programación estructurada Dev-C++**
* **Manuel de prácticas FO-aca**

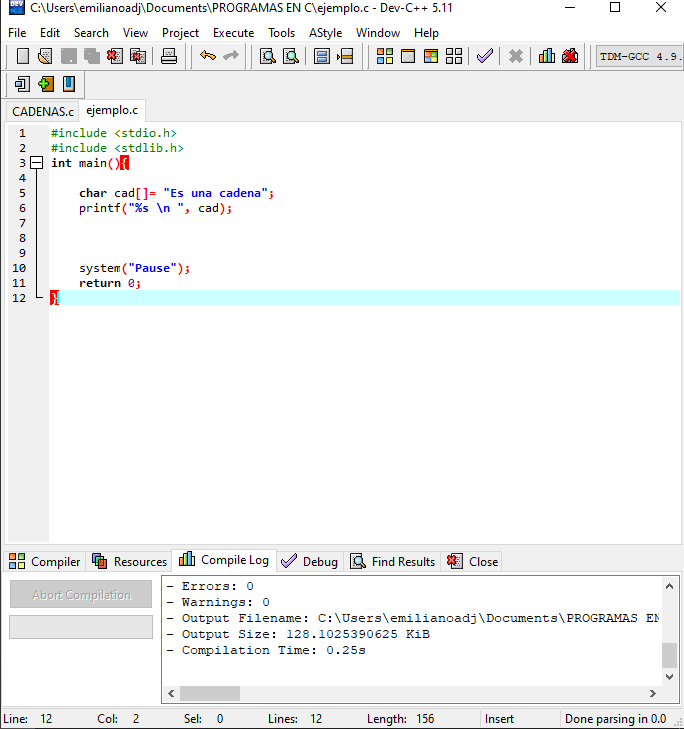
1. **Desarrollo de la práctica:**

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

**Para declarar una cadena, es lo mismo como un arreglo: char cadena[n];**

**Ejemplo de cadena básico.**

**Básicamente lo que hace el programa es que pide al usuario ingrese un apalabra e imprime el nombre de la cadena con la palabra que fue ingresada (línea 9).**

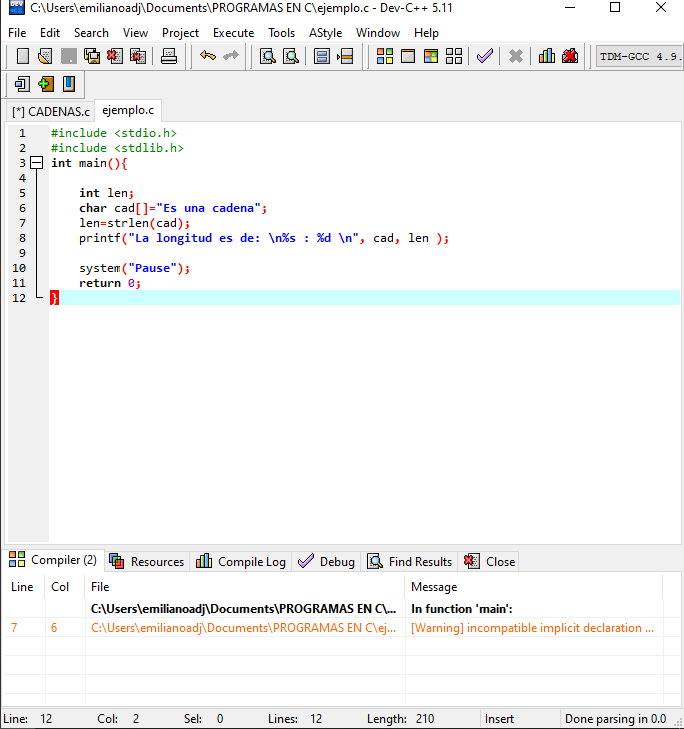
* **Si la definición de cadena dice que el ultimo carácter debe ser \0, entonces, ¿De cuántos elementos es la cadena “Es una cadena”? R= 14.**

**Solamente se está imprimiendo el contenido que ya se agrego manualmente a la cadena.**

**\*\*\*Función de cadena strlen.\*\*\***

**Esta función de cadenas devuelve el tamaño de una cadena en números enteros.**

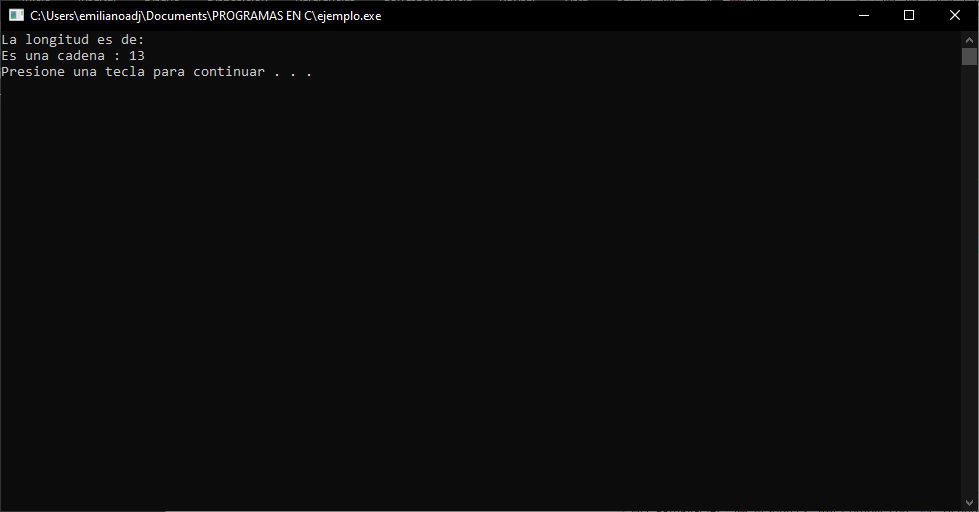
**strlen(cadena);**

* **Tamaño de una cadena**

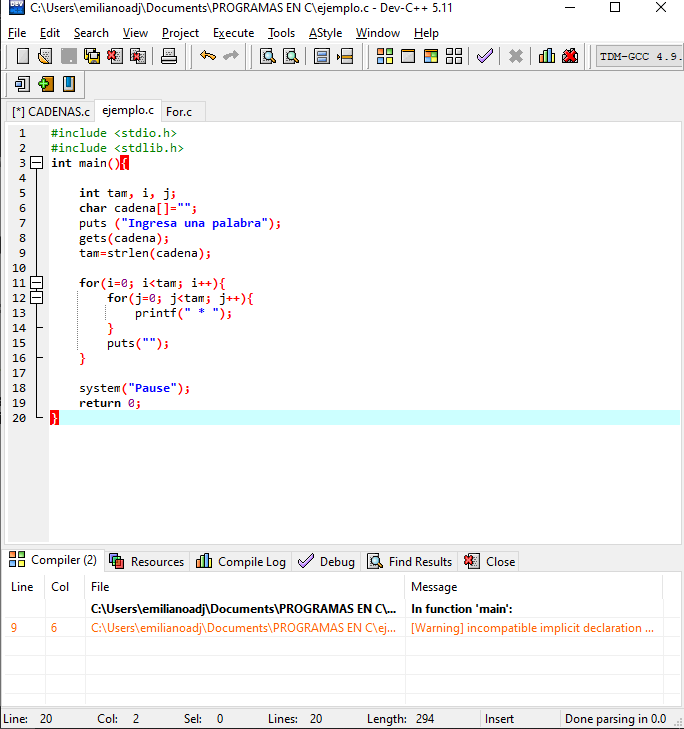
**Strlen devuelve el tamaño de una cadena en números enteros, de manera que el numero de la cadena se deberá guardar en una variable de tipo entero y así pueda mostrar la longitud.**

**Obtiene el tamaño (línea 7).**

**Muestra la cadena y el tamaño (línea 8). En este caso deberá mostrar 13.**



* **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).**



**En la variable llamada tam (tamaño) se guardo el tamaño entero de la cadena, de la palabra que ingreso el usuario.**

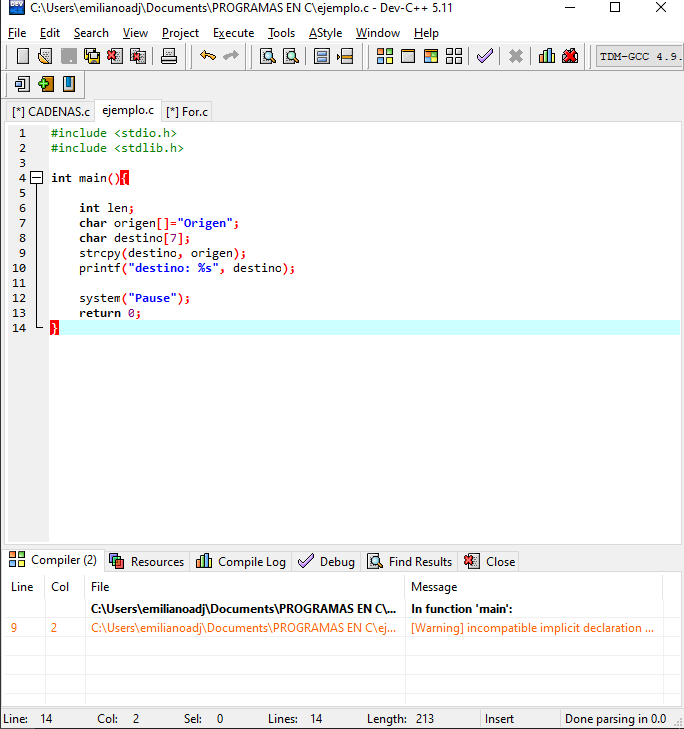
**Posteriormente con un ciclo for anidado vamos a imprimir con asteriscos un cuadrado de n\*m, es decir, que de altura tenga el tamaño de la cadena y de ancho tenga también el tamaño de la cadena.**

**\*\*\*Funciones de cadena strcpy\*\*\***

**Su función es copiar el contenido de una cadena origen a una cadena destino, es decir, que copia lo que hay en una cadena a otra.**

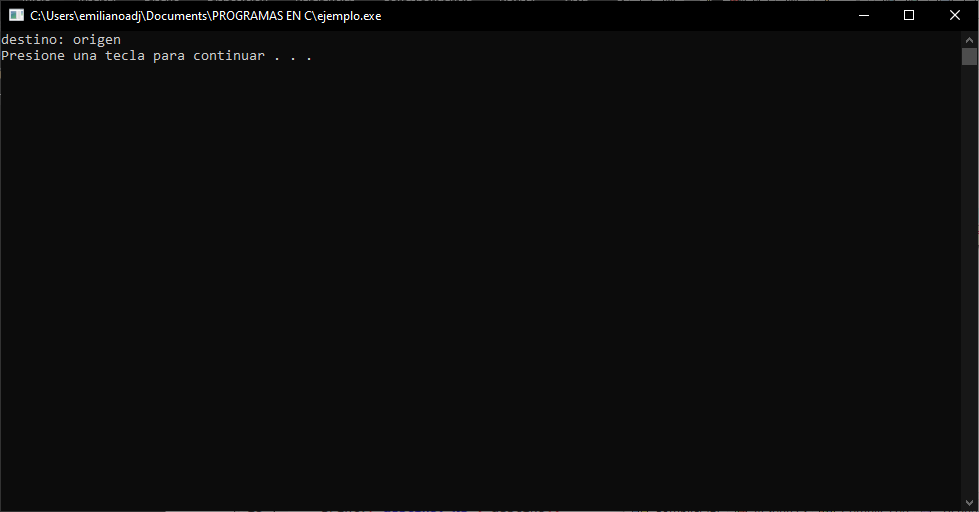
**Strcpy(<cadena\_destino> , <cadena\_origen>**

* **Ejemplo de strcpy:**

**Lo que hace el programa es copiar lo que hay en la cadena origen a destino.**

**(línea 9).**

**Imprime lo que fue copiado de origen, en destino.**



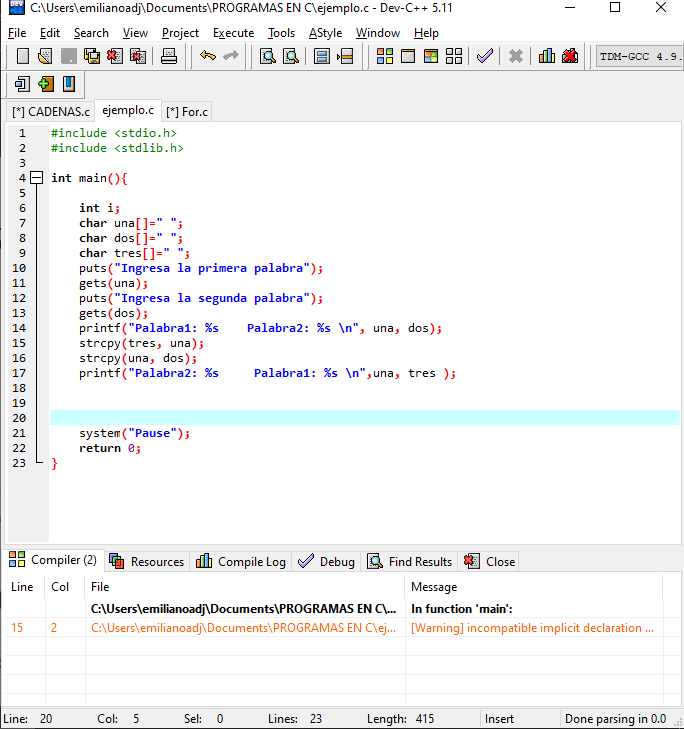
* **Escribe un programa que reciba por teclado dos palabra y cada una de ellas las almacene en un arreglo.**

**Después intercambia sus contenidos. Imprime el antes y el después. Ejemplo:**

**antes después**

**Palabra\_1 = ' Programación' Palabra\_1 = 'Computadora'**

**Palabra \_2= 'Computadora' Palabra \_2= 'Programación'**

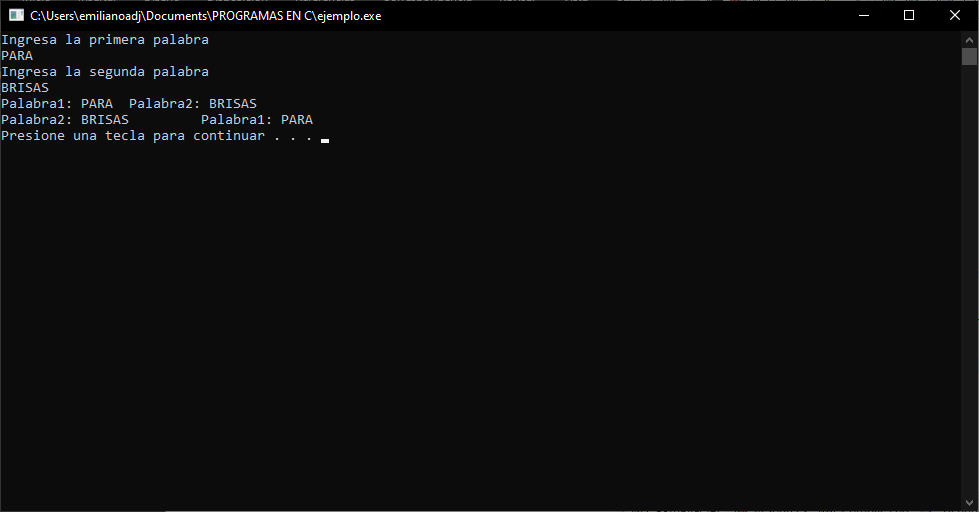


**El usuario ingresa dos palabras por separado, en una se guarda la palabra 1, en dos se guarda la palabra 2.**

**En tres vamos a copiar lo que contiene una.**

**En una copiaremos lo que contiene dos.**

**Para así poder imprimir como en el ejemplo.**

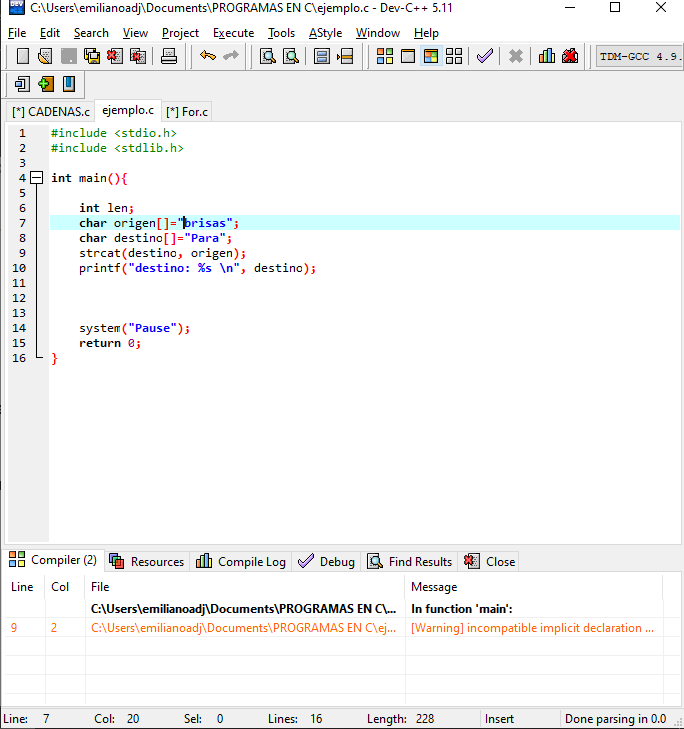


**\*\*\*Funciones de cadena strcat\*\*\***

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

**Estructura:**

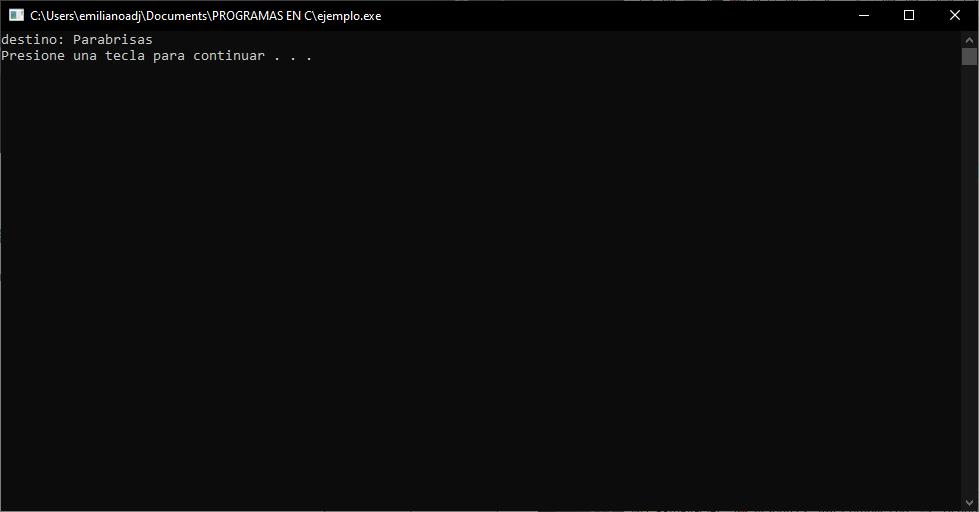
**strcat(<cadena\_destino>, <cadena\_origen>)**

* **Ejemplo de concatenar dos silabas para formar una palabra.**

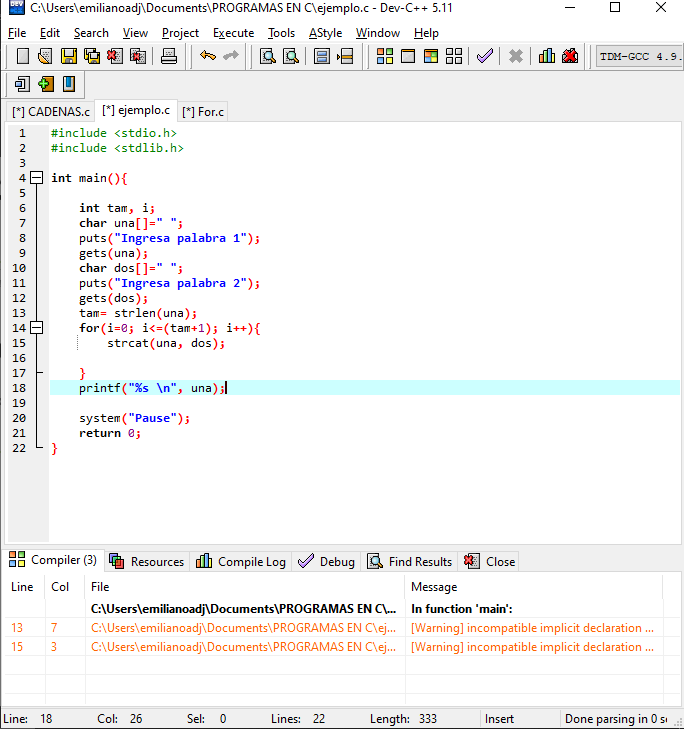
**En la cadena origen estamos escribiendo la sílaba “brisas”. (línea 7).**

**En la cadena destino estamos escribiendo la primera silaba “Para” (línea 8).**

**Cuando se utiliza strcat va a juntar las dos silabas en la cadena destino creando “Parabrisas” (línea 9-10).**



* **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.**

 **Ejemplo:**

**Si la primera palabra ingresada es perro, debe obtenerse el tamaño de esa palabra (5).**

**Supongamos que la segunda palabra es salchicha, entonces debe concatenarse a perro 5 veces salchicha:**

**Perro**salchicha**salchicha**salchicha**salchicha**salchicha.

**\*\*\*Funciones de cadena strcmp\*\*\***

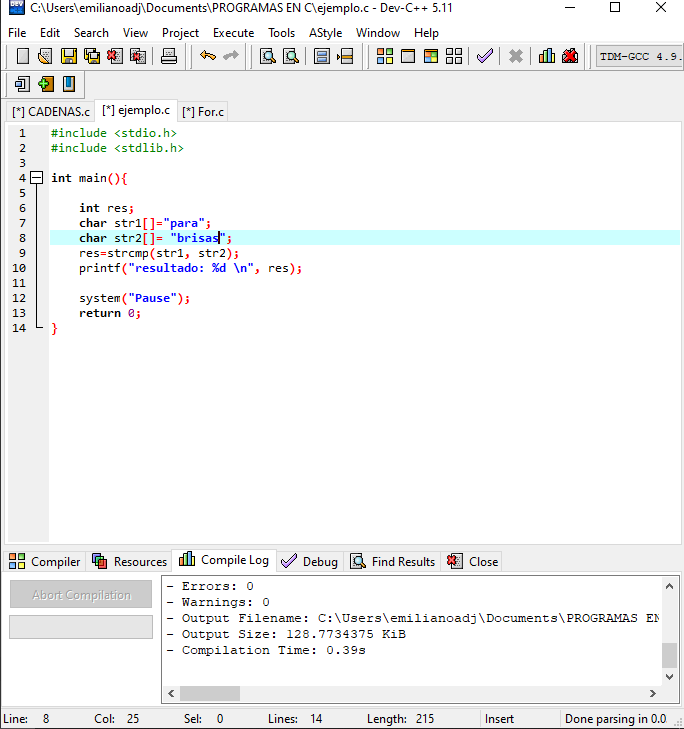
**Compara las cadenas y devuelve 0 si las cadenas son iguales.**

**Un numero positivo mayor que 0 si <cadena1> es menor a la <cadena2>**

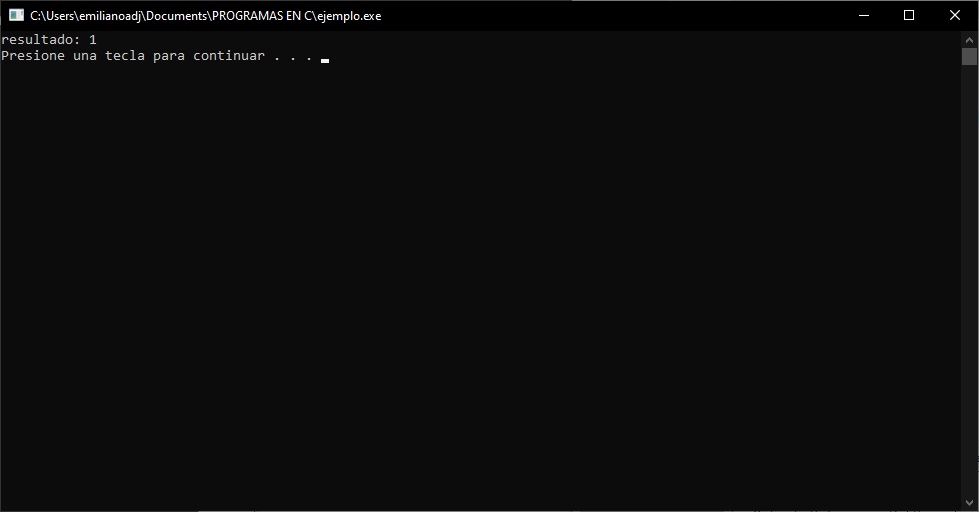
**Un numero negativo si la <cadena1> es mayor que la <cadena2>**

**Sintaxis: strcmp(<cadena1>, <cadena2>),**

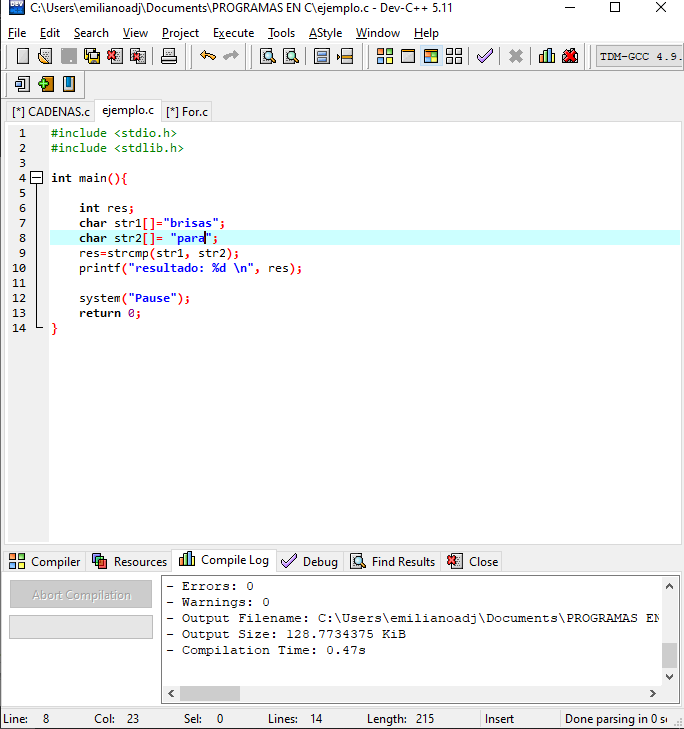
* **Ejemplo**

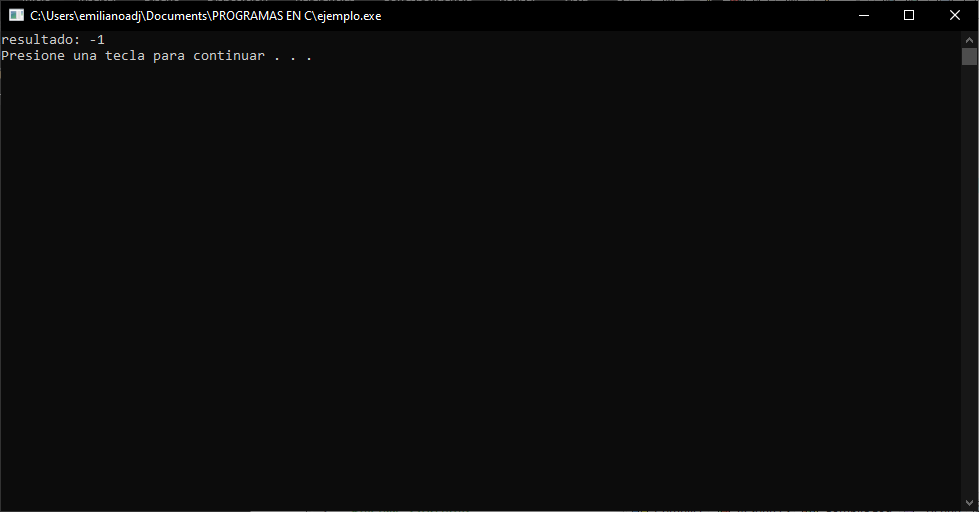


**Tomando en cuenta las especificaciones antes mencionadas, este programa debería imprimir un numero 1.**



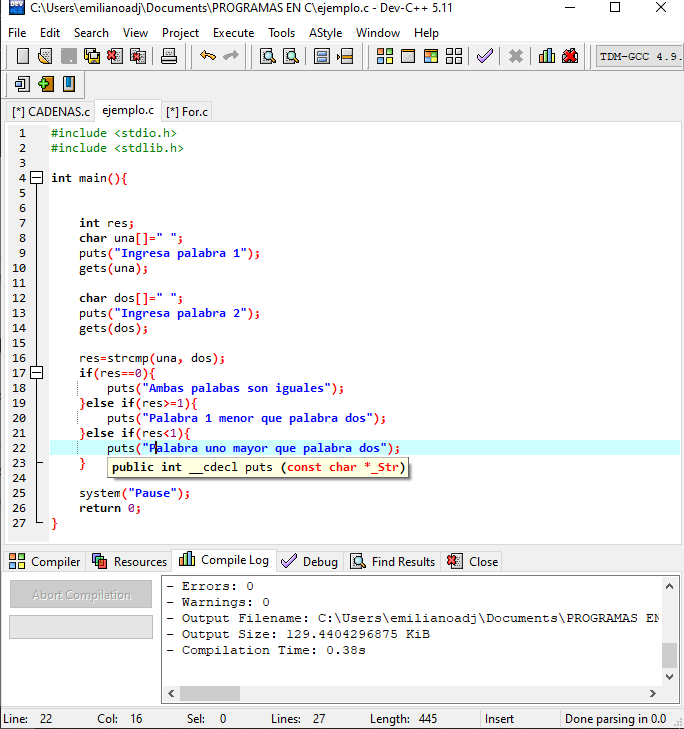
**Esto pasaria silas palabras estuviesen en orden inverso, si para estuviera abajo y brisas arriba.**

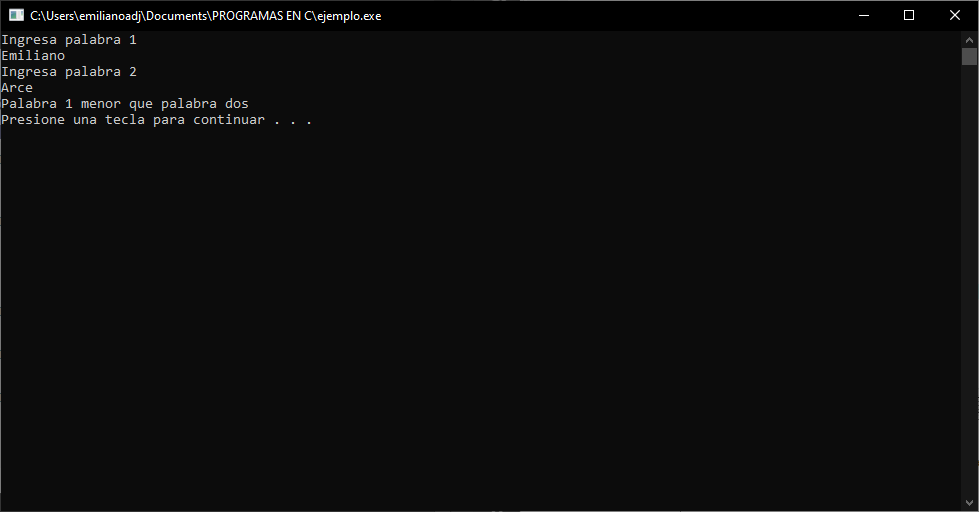


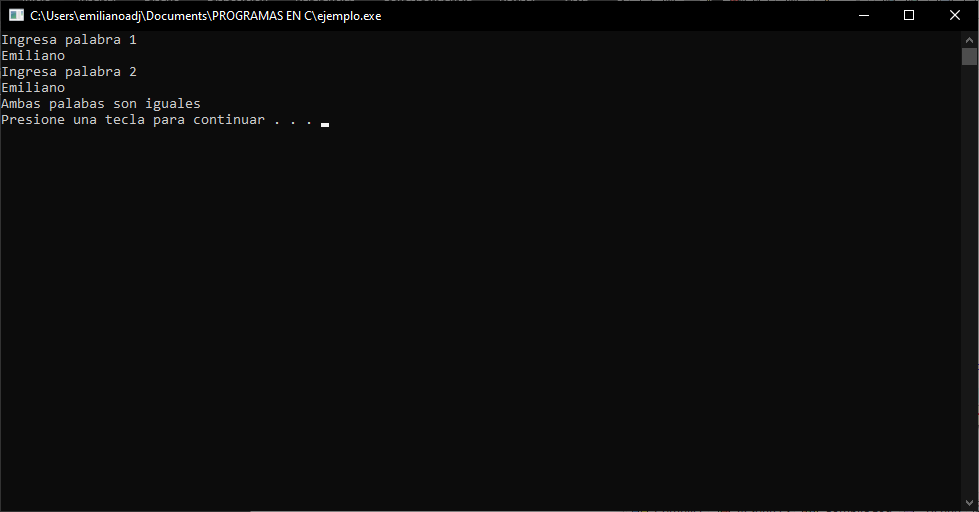
**El resultado debe ser -1.**

* **Escribe un programa que reciba dos palabras y te indique cual de ellas es menor y cual de ellas es mayor.**

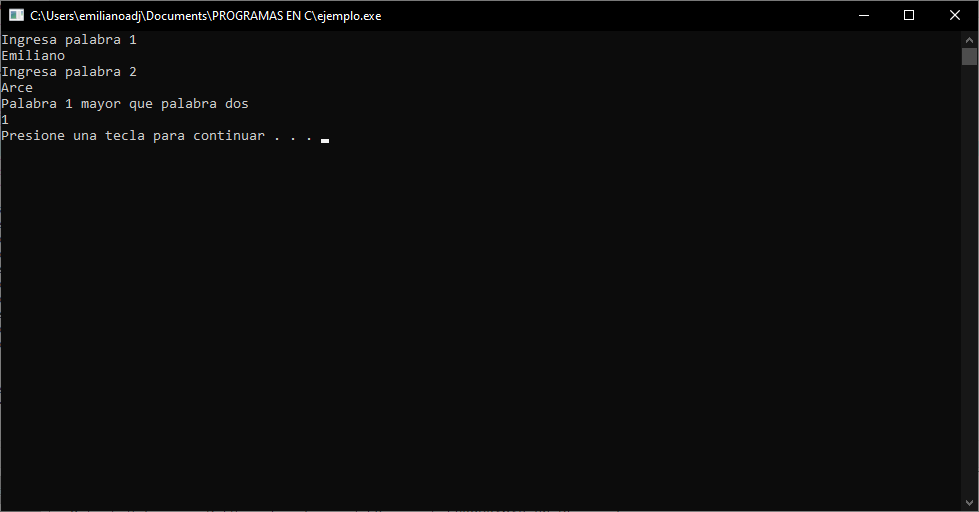
**En caso de ser iguales que imprima la leyenda: ”Ambas palabras son iguales”.**



**Introducimos 1: Arce 2: Emiliano res= -1 : “Palabra 1 mayor que palabra 2”.**



**Introducimos 1: Emiliano 2: Emiliano res: 0 : “Ambas palabras son iguales”.**

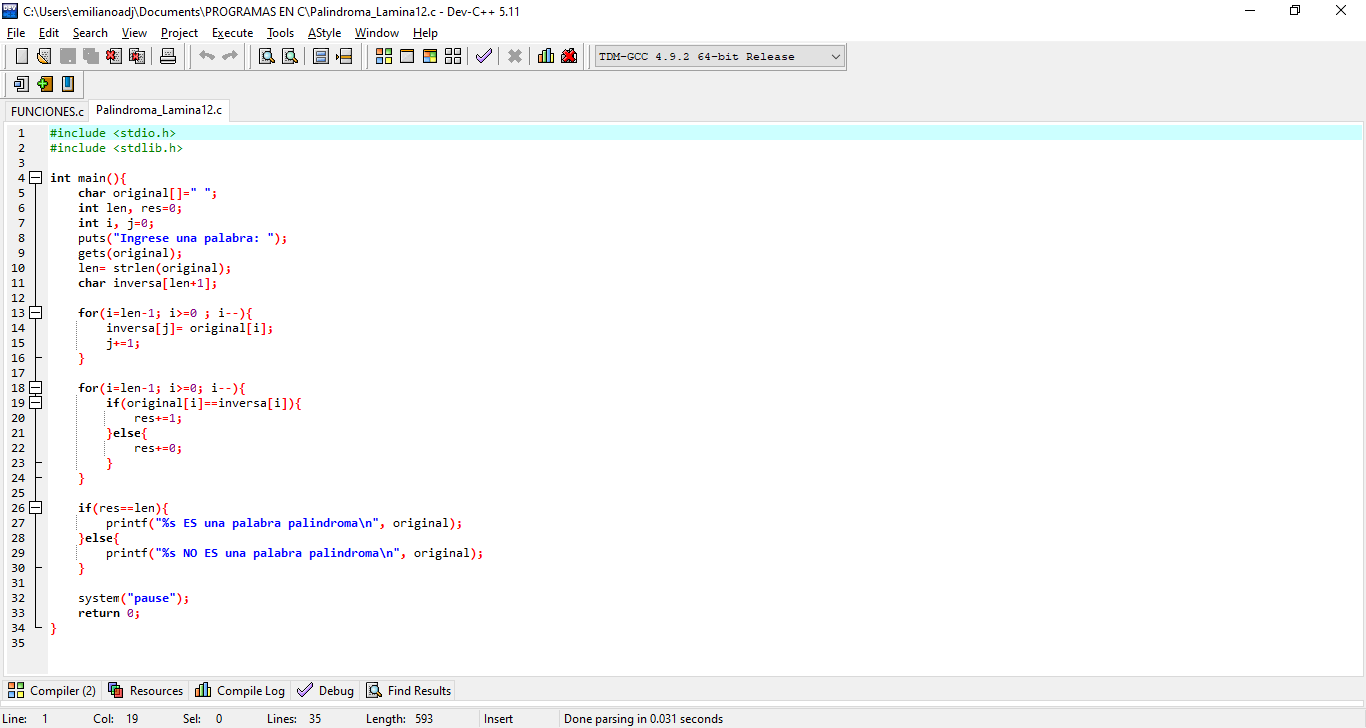
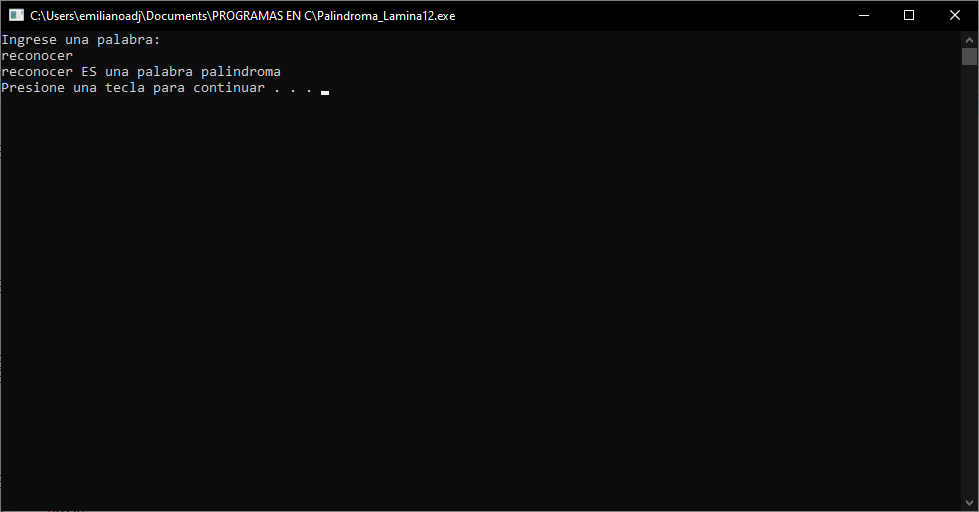


**Introducimos 1: Emiliano 2: Arce res= 1 : “Palabra uno menor que palabra 2”.**

* **Crea un programa que detecte una palabra palíndroma. Las palabras palíndromas son palabras que se leen igual de derecha a izquierda como de izquierda a derecha.**

**Se hace un proceso de intercambio de posiciones, se guarda en otro arreglo la palabra ingresada, pero al revés, y después con un conjunto de estructura for e if se evalúa si el contenido de las posiciones de cada arreglo es meramente igual, si es así, habrá un contador y se acumulará.**

**Finalmente se comparan el contador con la cadena original y se puede deducir si la palabra es palíndroma o no.**



**\*\*\*Funciones\*\*\***

**En programación, una función es un bloque de código reconocido por un identificador que realiza un trabajo específico. Su función es dividir s los programas en módulos manejables separados (divide y vencerás).**

**Declaración de una función:**

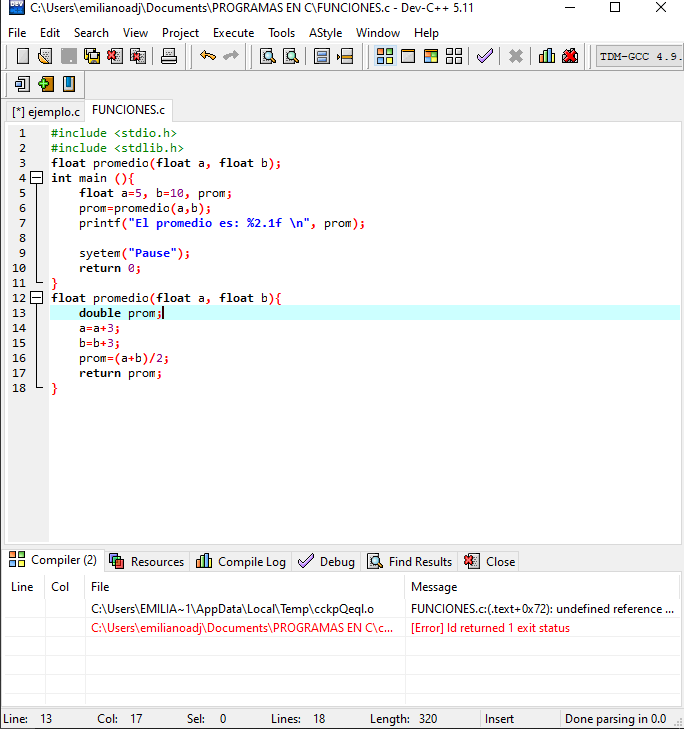
**tipo nombreFuncion(lista de parámetros) {**

**cuerpo de la función**

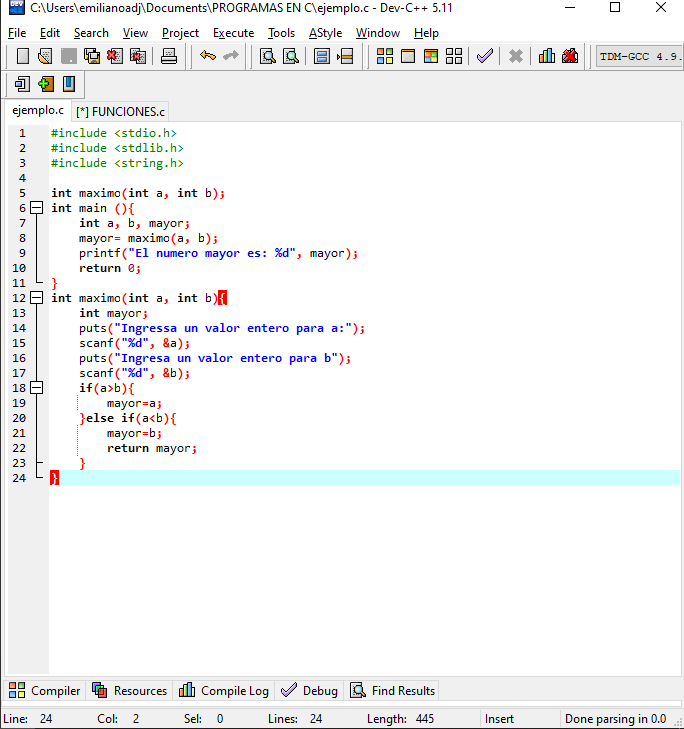
**}**

* **Ejemplo de un función.**

**En el siguiente programa se trata de realizar una operación con los parámetros establecidos.**

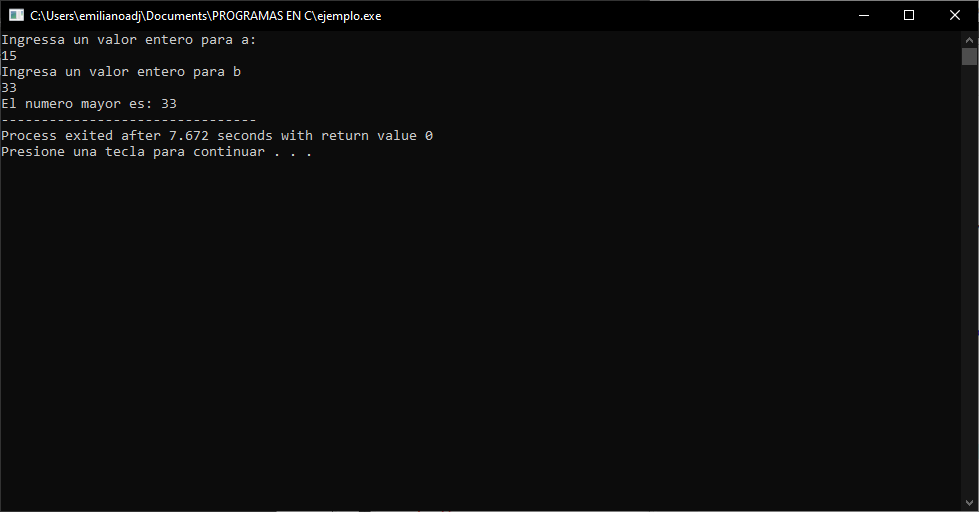


* **Escribe una función que se llame máximo que reciba dos números por parámetros y que regrese el mayor de ellos.**

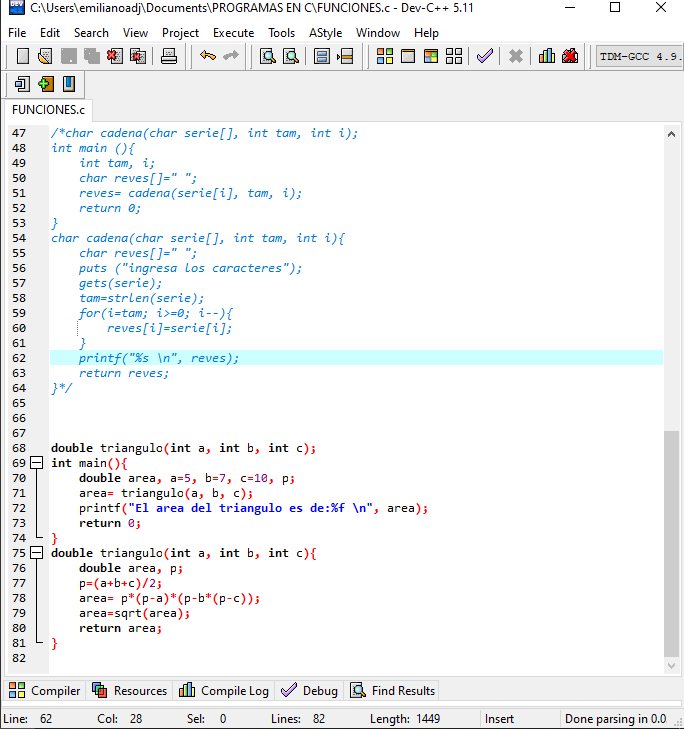


**Cuando se ingresan dos números (a, b) el programa hace una comparacion en el cuerpo de la funcion verificando cual es el numero mayor y lo guarda y retorna en la variable mayor (línea 18-22)**

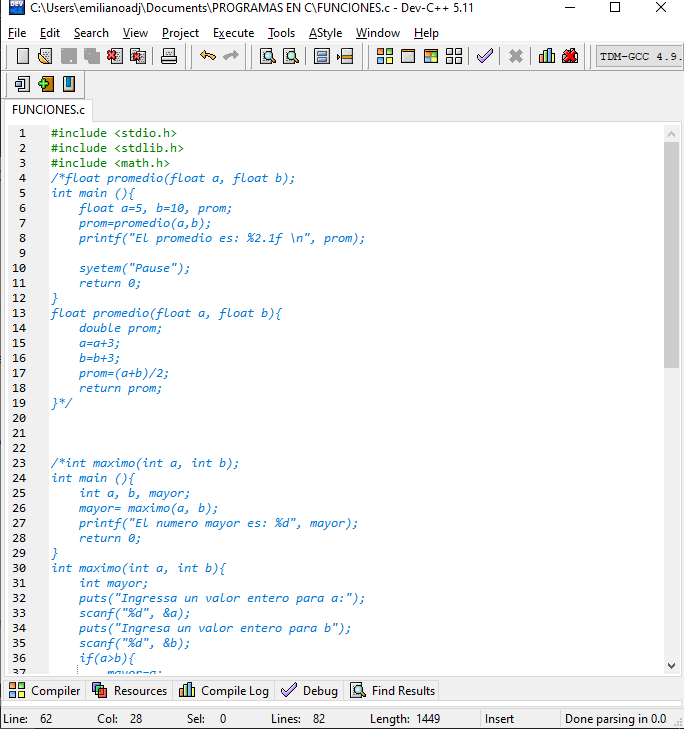
**Imprimiendo finalmente el numero mayor ingresado.**



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



**Para sacar una raíz cuadrada, primeramente debimos importar la biblioteca <math.h>**

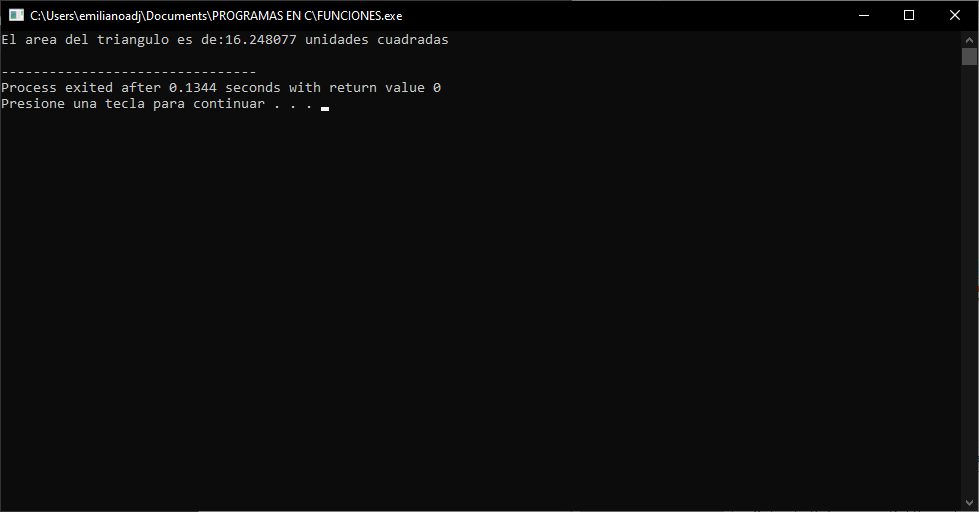


**Debidamente procedemos a utilizar la formula**

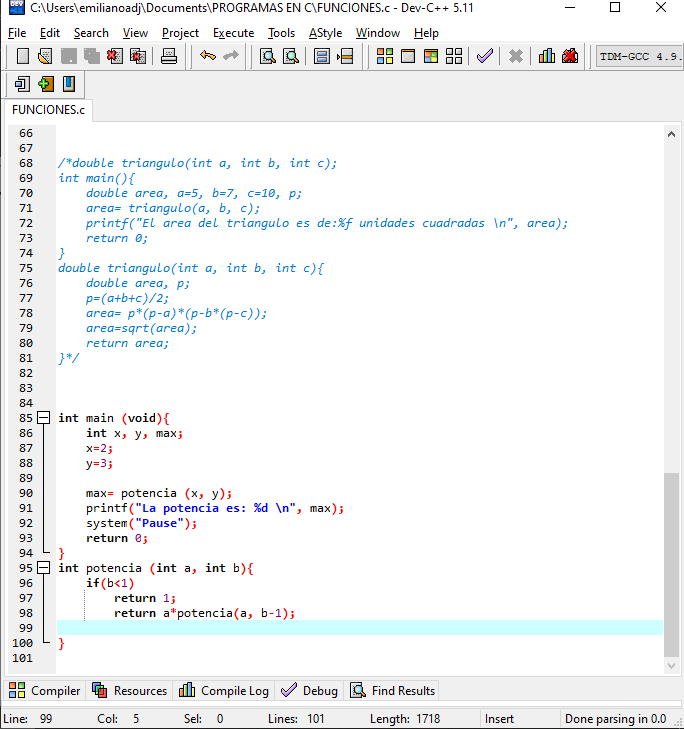
**La cual nos ayudara para realizar bien nuestra operación.**

**Realizamos primero lo de adentro de la raíz para finalizar realizando la raíz cuadrada (línea 77-79).**

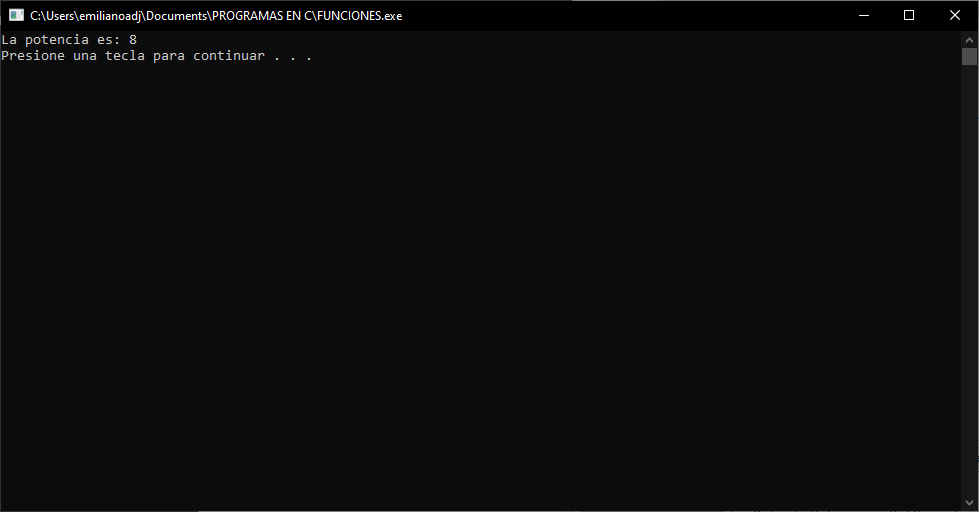
**Finalmente se retorna área y se imprime (línea 71-72).**

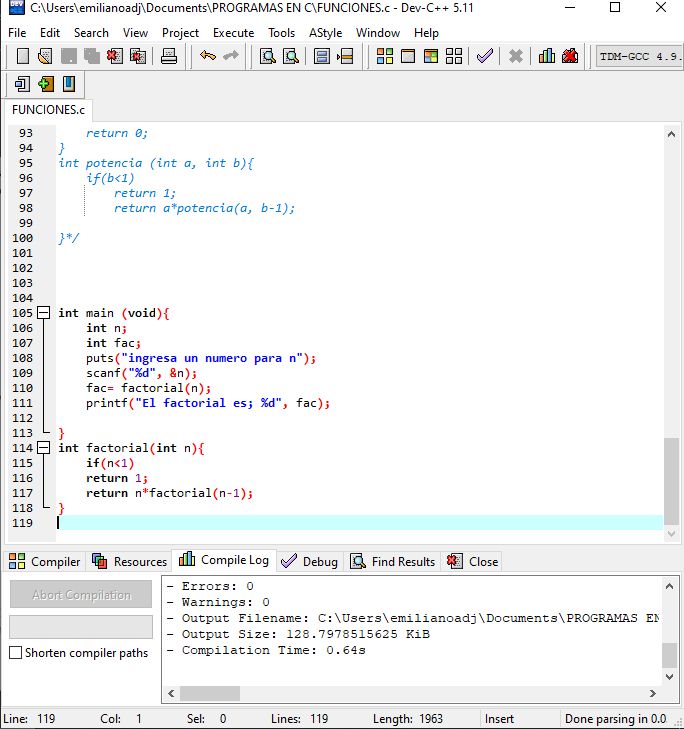


* **El siguiente ejemplo es para obtener la potencia de dos datos establecidos.**



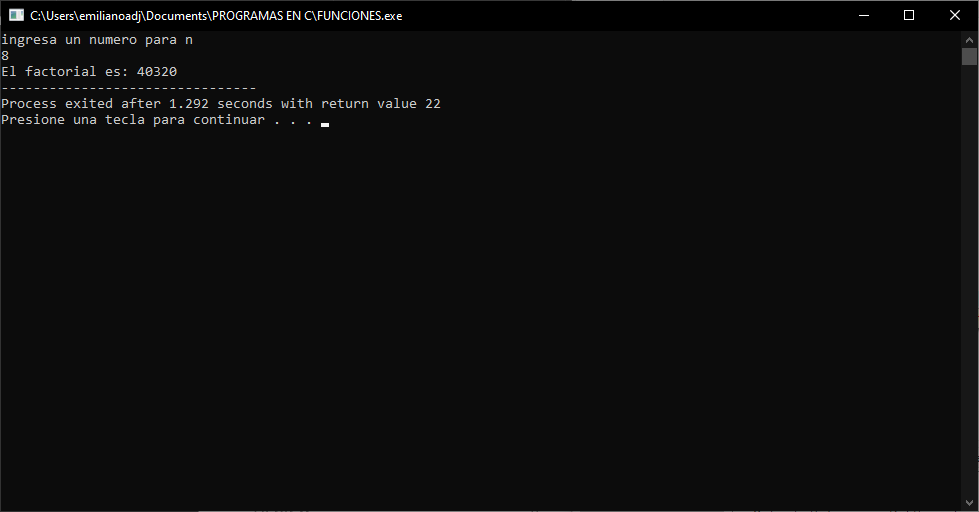
**Hay función con recursividad en la parte donde b debe ser menor a 1, en este caso se va a mandar llamar varias veces la misma función hasta que se cumpla la función.**





* **Has un programa con funciones recursivas que calcule la factorial de un numero n ingresado desde teclado.**

**Una factorial es una multiplicación consecutiva de 0**

**hasta n.**

1. **CONCLUSIONES:**

**Cadenas tiene una amplia área de ejercicios con los que se pueden resolver problemas sencillos como por ejemplo verificar datos repetidos entre otros.**

**Funciones, las funciones hacen un programa mas entendible y mas ordenado.**

**Con esto se logra la que sea legible y pues de esta manera poder tener un mejor control de los programas que se vayan realizando.**