|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre de la práctica** | **PROGRAMAS CADENAS** | | | **No.** | **3** |
| **Asignatura:** | **METODOS NUMERICOS** | **Carrera:** | **INGENIERÍA EN SISTEMAS COMPUTACIONALES** | **Duración de la práctica (Hrs)** | **3 horas** |

**NOMBRE DEL ALUMNO: Alejandro Trejo Godinez**

**GRUPO: 3042**

**I. Competencia(s) específica(s):**

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

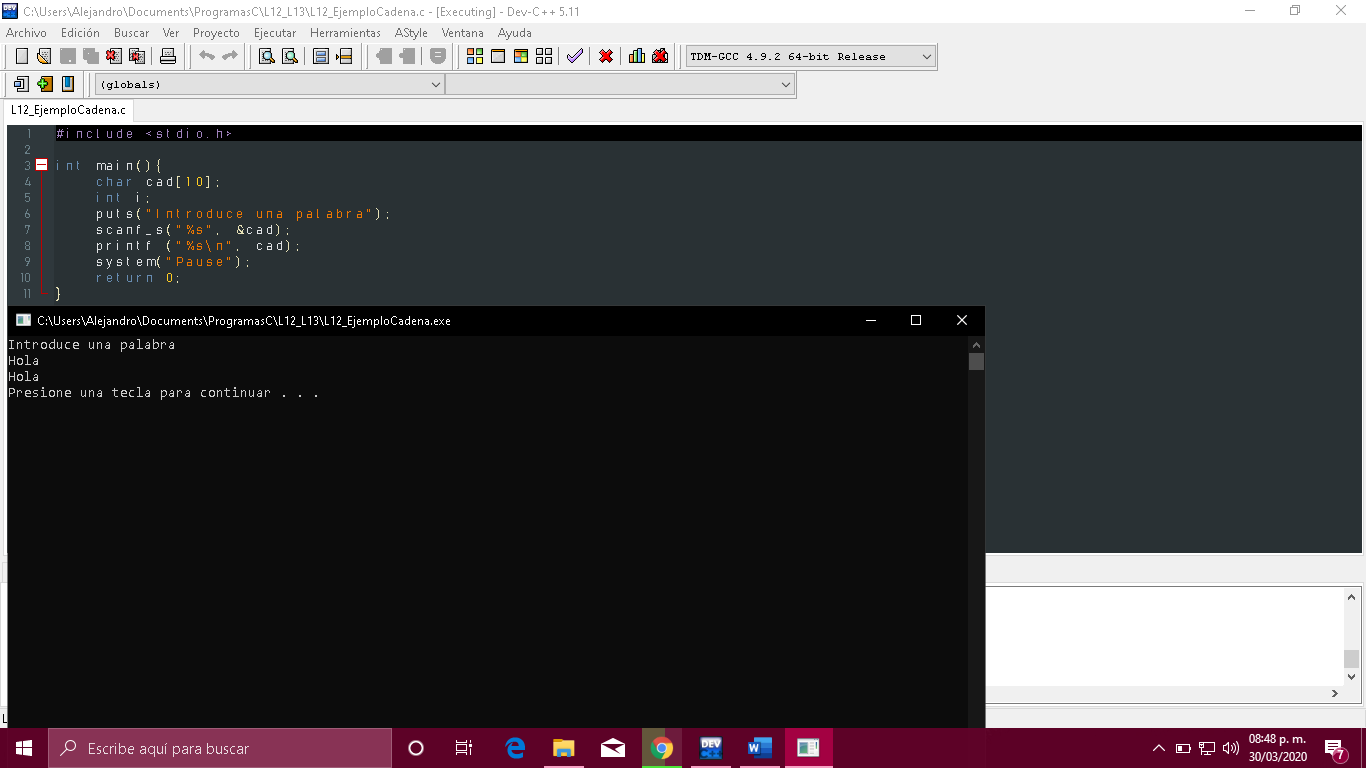
Aula de clases

**III. Material empleado:**

* Equipo de cómputo
* Dev C++

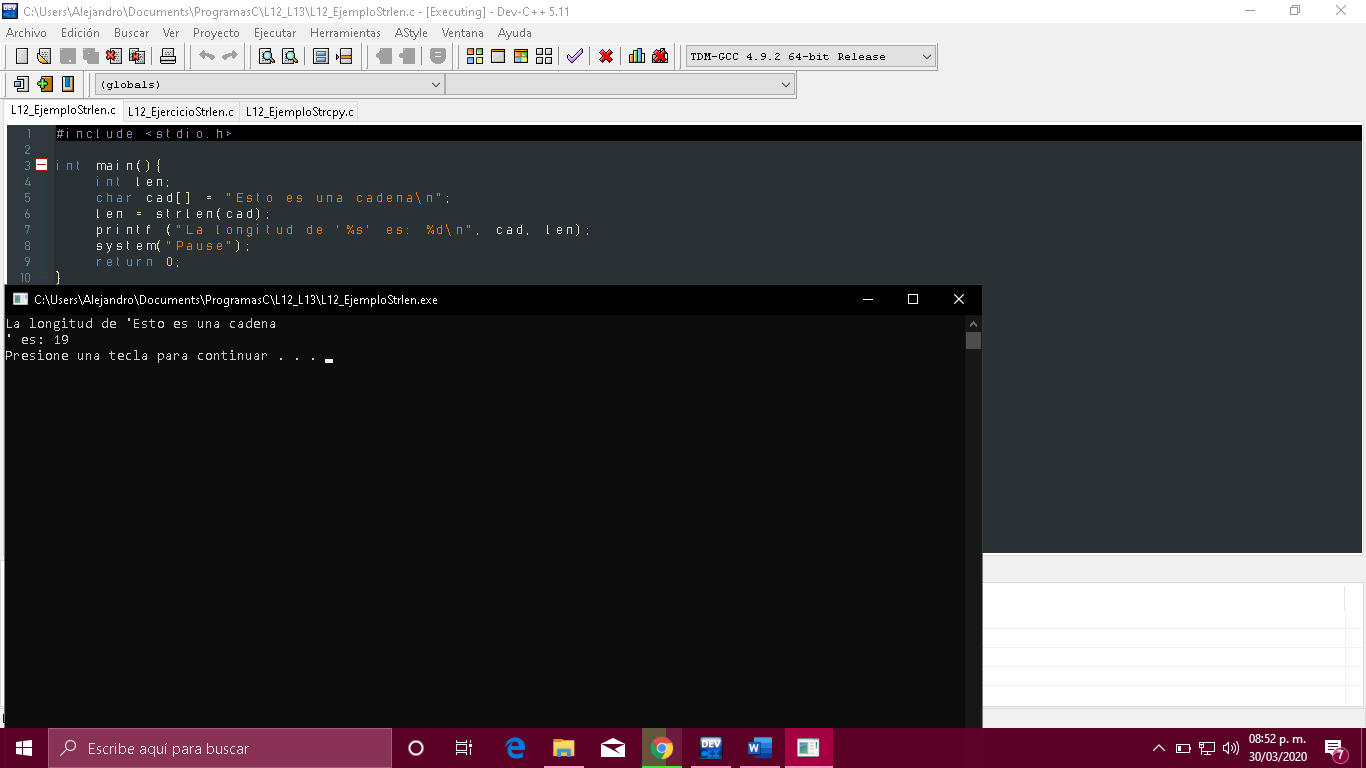
**IV. Desarrollo de la práctica:**

# Ejemplo Cadena

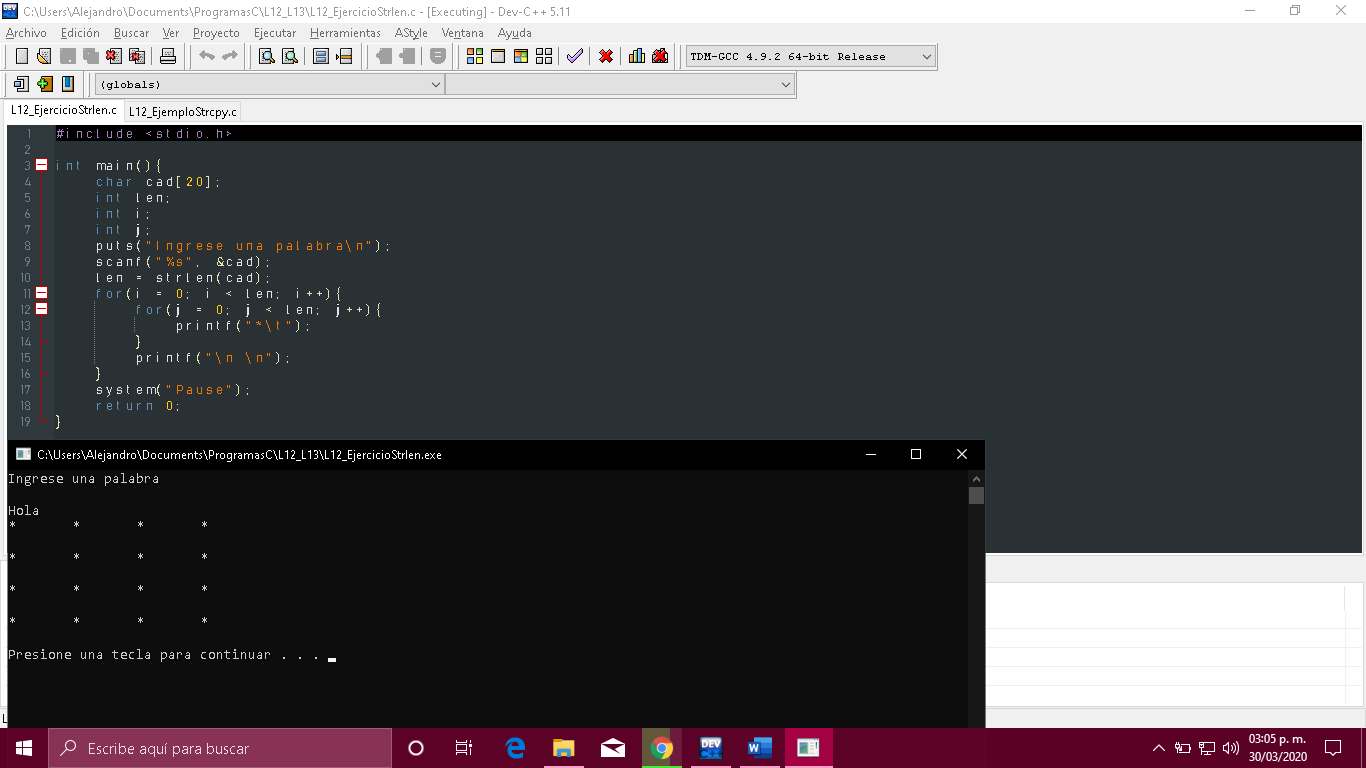


En este primer ejemplo sobre cadenas, descubrimos como almacenar dichas cadenas, se declara el tipo de dato **char** y se le da el tamaño a la cadena, posteriormente se imprime un mensaje en el que pide se introduzca una palabra, al ingresarla se almacena en la variable de tipo char y posteriormente se imprime.

# Ejemplo strlen

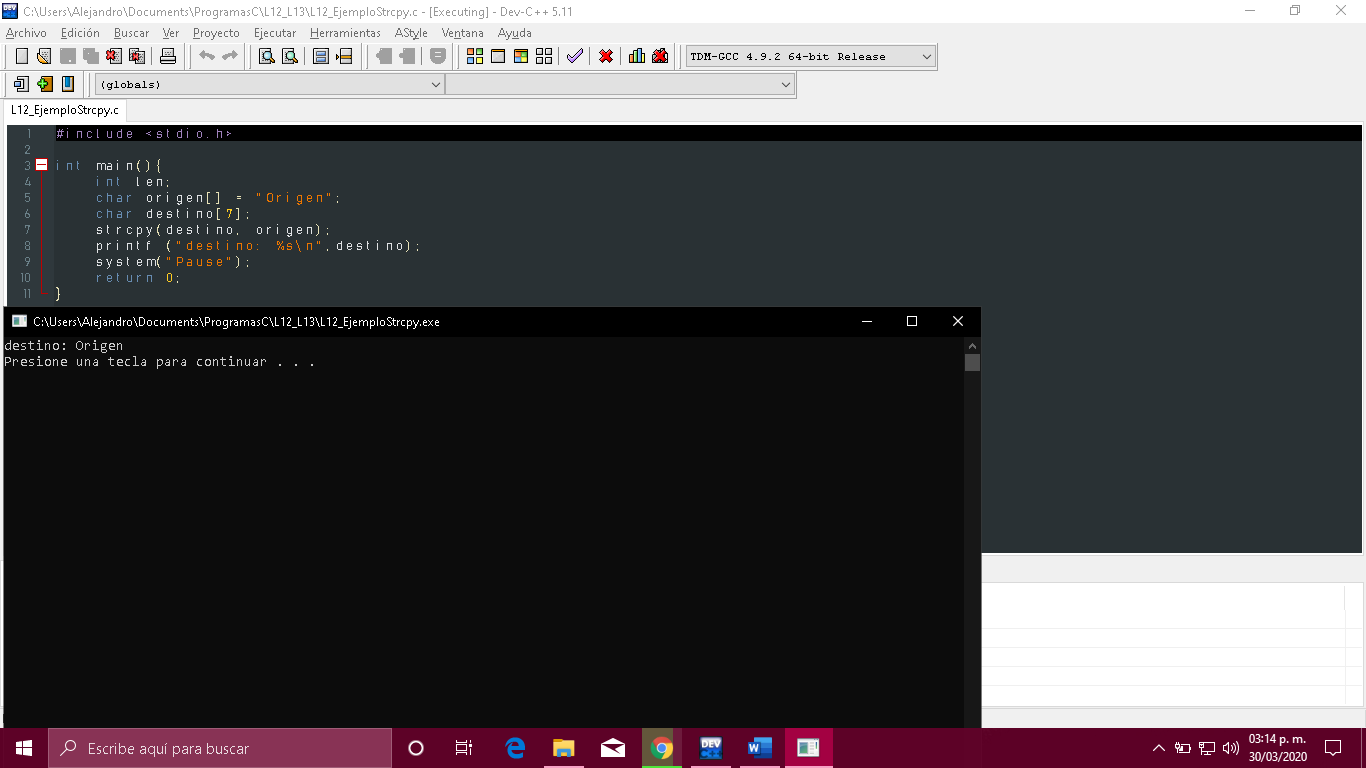
En este programa aprendemos el uso de la instrucción **strlen** la cual nos devuelve el tamaño de la cadena que le indiquemos, en este ejemplo e declara una variable de tipo int, se declara la cadena y posteriormente se obtiene su tamañano con strlen y se almacena el resultado en la variable de tipo int, imprimimos el mensaje donde mostramos la cadena y el tamaño de esta.

# Ejercicio strlen



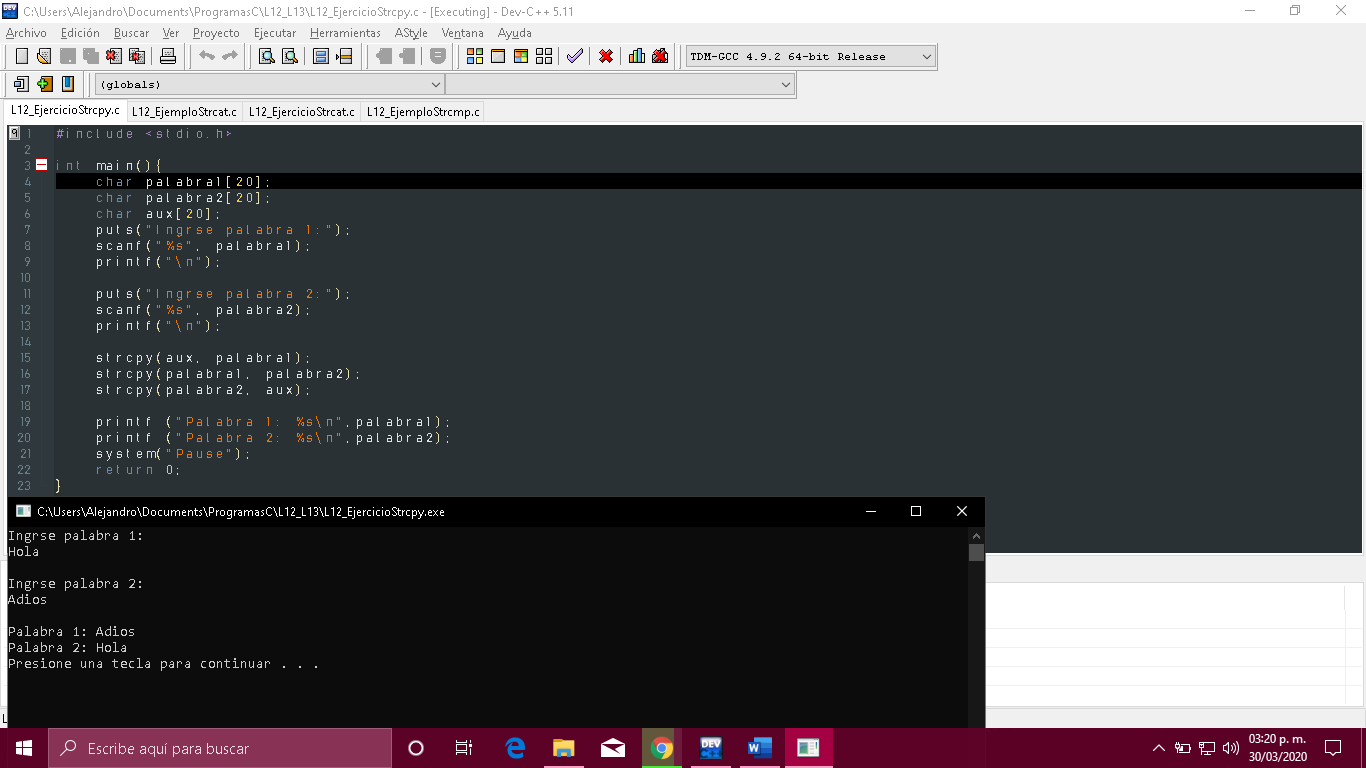
En este ejercicio, se nos pidió que en base a una palabra ingresada se genere una matriz de asteriscos de nxn, siendo n la longitud de la cadena ingresada. Primero se crea una variable de tipo char donde almacenaremos la cadena, tres de tipo int, una para almacenar la longitud de la cadena y las otras dos nos servirán de índices para los ciclos for. Se ingresa la cadena y se obtiene su longitud, ingresamos al primer ciclo for y la condición dice que saldrá del ciclo cuando el índice sea igual a la longitud de la cadena, entra a otro for con la misma condición y dentro de este último, es donde imprime el asterisco, sale de este for anidado y da el salto de línea y lo hará asi cada que deba terminar una línea de asteriscos.

# Ejemplo strcpy



En este ejemplo aprendemos el uso del **strcpy** el cual sirve para copiar cadenas. Declaramos nuestra cadena origen y le colocamos la cadena de caracteres, posteriormente declaramos otro pero solo le asignamos el tamaño, despues hacemos uso de la instrucción **strcpy** cuya sintaxis es **strcpy(cadenadestino, cadenaACopiar);** de esta manera copia lo ue hay en una cadena hacia otra, finalmente solo se imprime el resultado de esta instrucción.

# Ejercicio strcpy

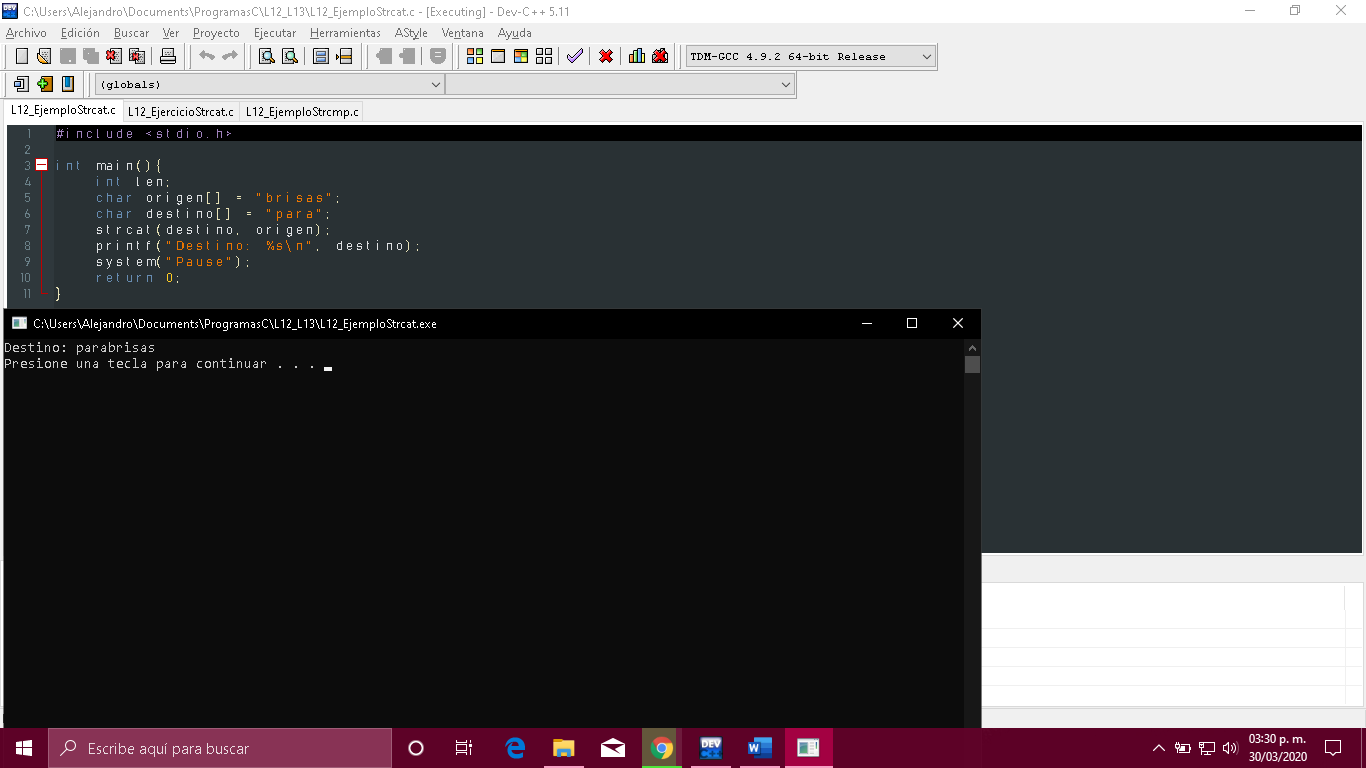


En este ejercicio declaramos dos variables de tipo char y les damos longitud de 20, declaramos una tercera que nos servira de apoyo.

Posteriormente pedimos que se ingresen las dos palabras y las almacenamos en nuestras variables, para mostrar priemero la ultima palabra y viceversa, nos apoyaremos de la tercer variable declarada, copiamos a esta lo almacenado en palabra 1, a palabra 1 le copiamos la almacenado en palabra 2, y a esta ultima lo almacenado en nuestro auxiliar.

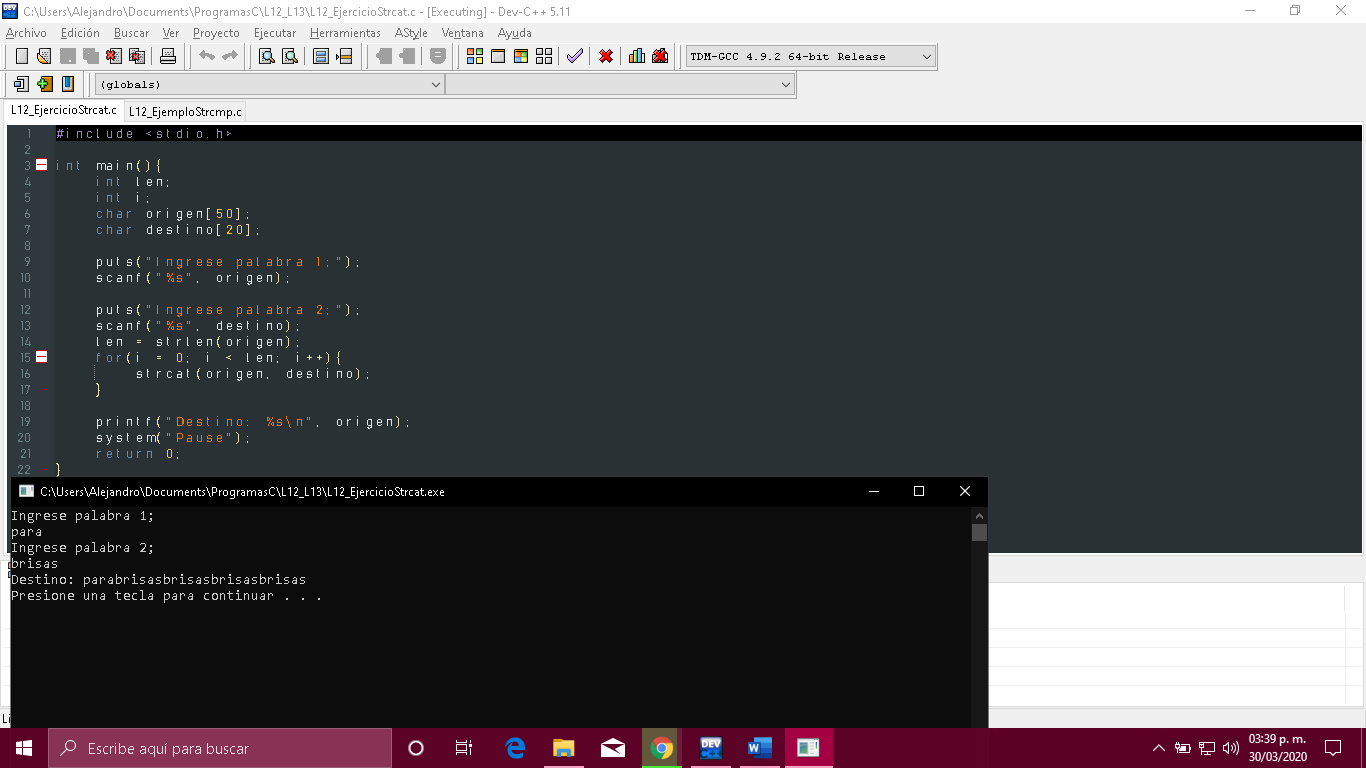
Al final solo imprimimos el resultado y veremos que efectivamente nos da el resultado esperado.

# Ejemplo strcat



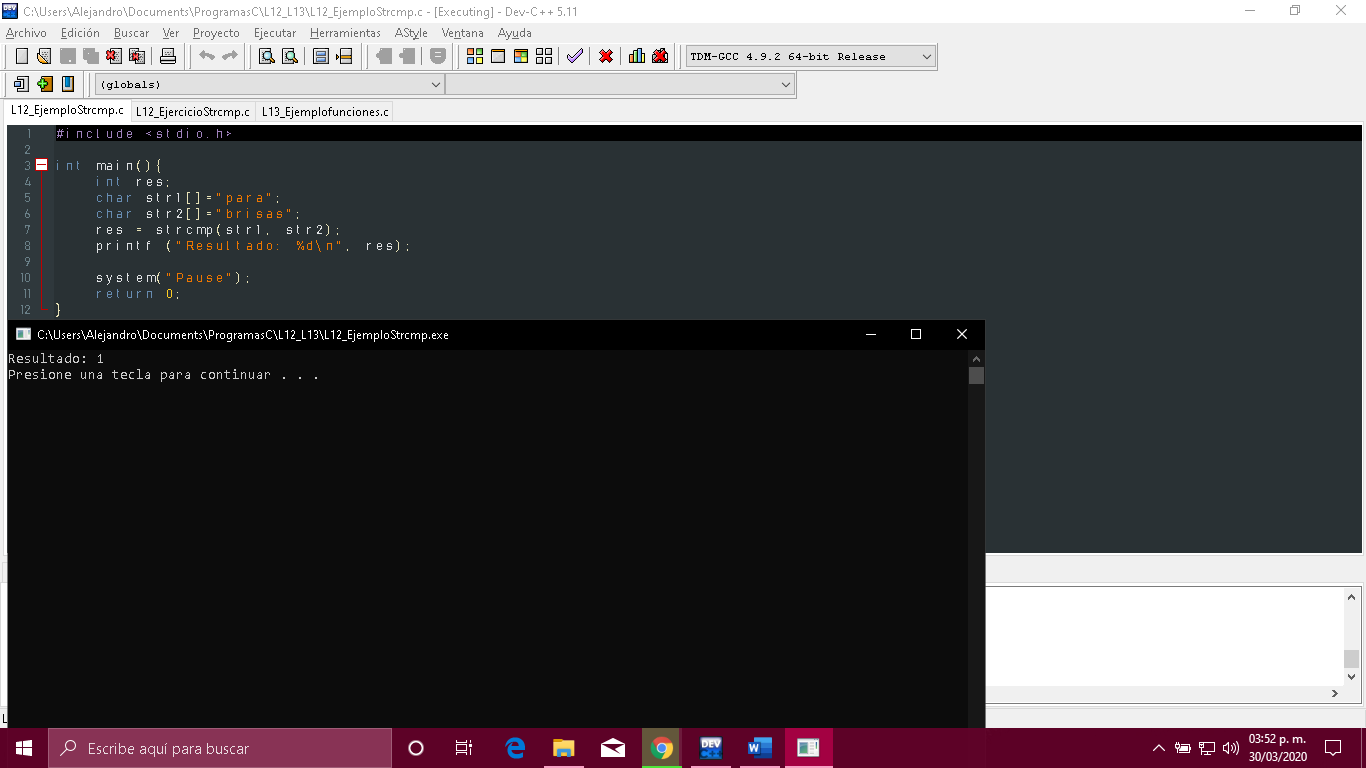
Ahora pasamos a otra instrucción relacionada a las cadenas, esta vez con strcat, la cual nos sirve sencillamente para concatenar cadenas, en este ejemplo declaramos dos cadenas y les asignamos sus respectivas cadenas, posteriormente ejecutamos la instrucción strcat, la cual tiene la siguiente sintaxis **strcat(cadena1, cadena2);** siendo cadena2 la que se concatenara en cadena1, al final imprimimos el resultado de la instrucción.

# Ejercicio strcat



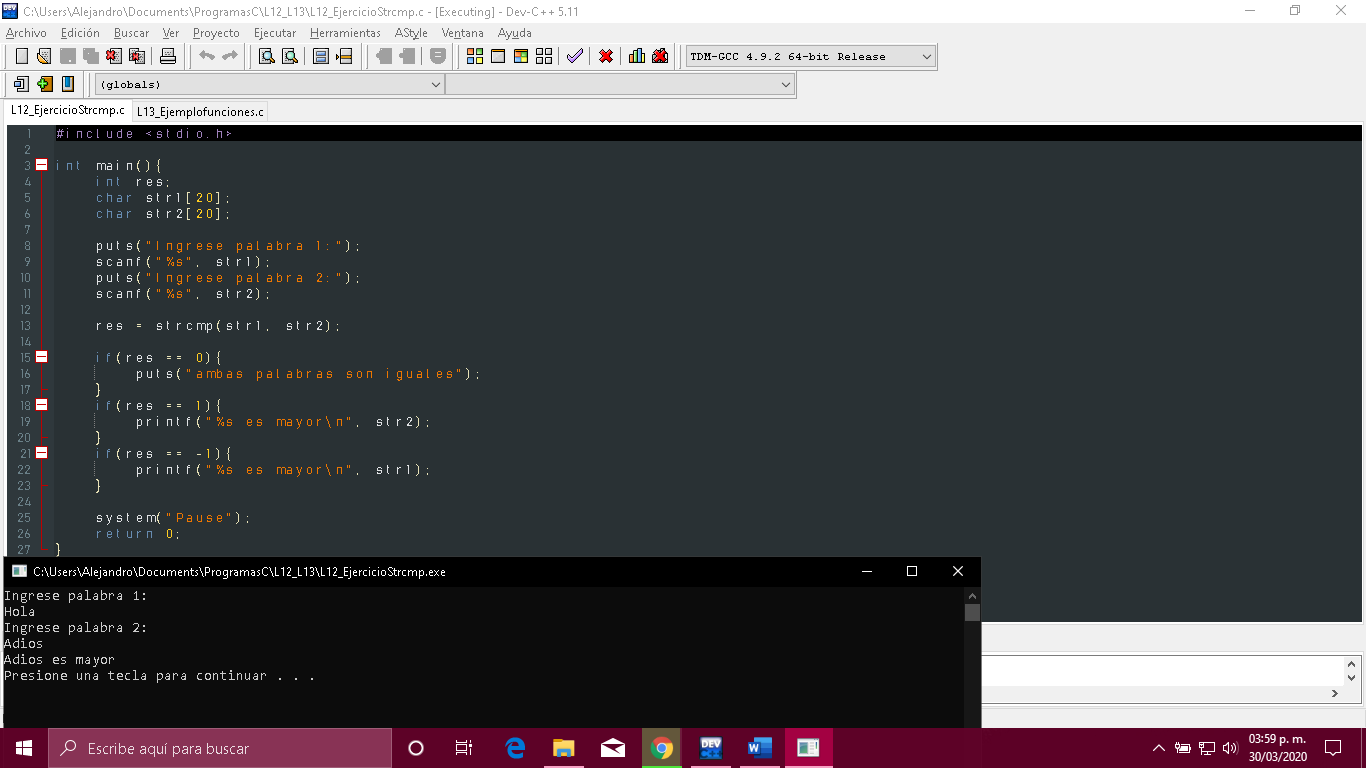
En este ejercicio, se nos pide que se concatene **n** veces la segunda palabra ingresada a la primera, siendo **n** la longitud de la primer palabra. Primero se declaran dos variables de tipo int, la primera nos sirve para almacenar el tamaño de la cadena y la segunda para recorrer el for. Despues declaramos dos te tipo char asignandoles el tamaño, por obviedad la que almacene la palabra 1 debe ser mucho mayor a la que almacene a la palabra 2, ingresamos las palabras, sacamos la longitud de la primer palabra y entramos al for, dentro de este usamos la instrucción strcat y se concatenara la palabra 2 en la 1 las veces que se repita el ciclo, al final imprimimos el resultado con un printf.

# Ejemplo strcmp



En este ejemplo aprendemos el uso del **strcmp** el cual se usa: **nombreVariable = strcmp(str1, str2);** donde nombreVariable es donde se almacenara el resultado de la comparación ya que strcmp devuelve un int (1 si la primer cadena es menor que la segunda, -1 si la segunda es menor que la primera y 0 si son iguales), dentro de los paréntesis solo se comparan las cadenas. En este ejemplo la primera es menor que la segunda, es por eso que nos imprime 1.

# Ejercicio strcmp



En este ejercicio, se nos pide que comparemos dos palabras y digamos cual es mayor, y en caso de ser iguales imprimir un mensaje donde lo diga. Primero declaramos una varible int donde almacenaremos el resultado de strcmp y dos de tipo char donde estaran las cadenas, las ingresamos por teclado y las comparamos, el resultado lo almacenamos en nuestra variable int, poteriormente la evaluamos, entra a un primer if donde preguntamos si el resultad fue 0, de ser asi nos dice que son iguales, si no pasa al siguiente if, si es 1 entonces imprime que el segundo es mayor, si no pasa al ultimo if donde si es -1 el primero es el mayor.