|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre de la práctica** | **Cadenas & Funciones en C** | | | **No.** | **11-12** |
| **Asignatura:** | **Métodos Numéricos** | **Carrera:** | **Ingeniería en Sistemas Computacionales** | **Duración de la práctica (Hrs)** | **1hra** |

**I. Competencia(s) específica(s)****;**

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

**III. Material empleado:**

* Dev-C++
* Libreta
* Word

**IV. Desarrollo de la práctica:**

**12. Cadenas**

Antes de empezar con los ejercicios hay que saber que es una cadena: es un arreglo de caracteres. En donde, por lo general el último elemento deberá ser el carácter ‘\0’.

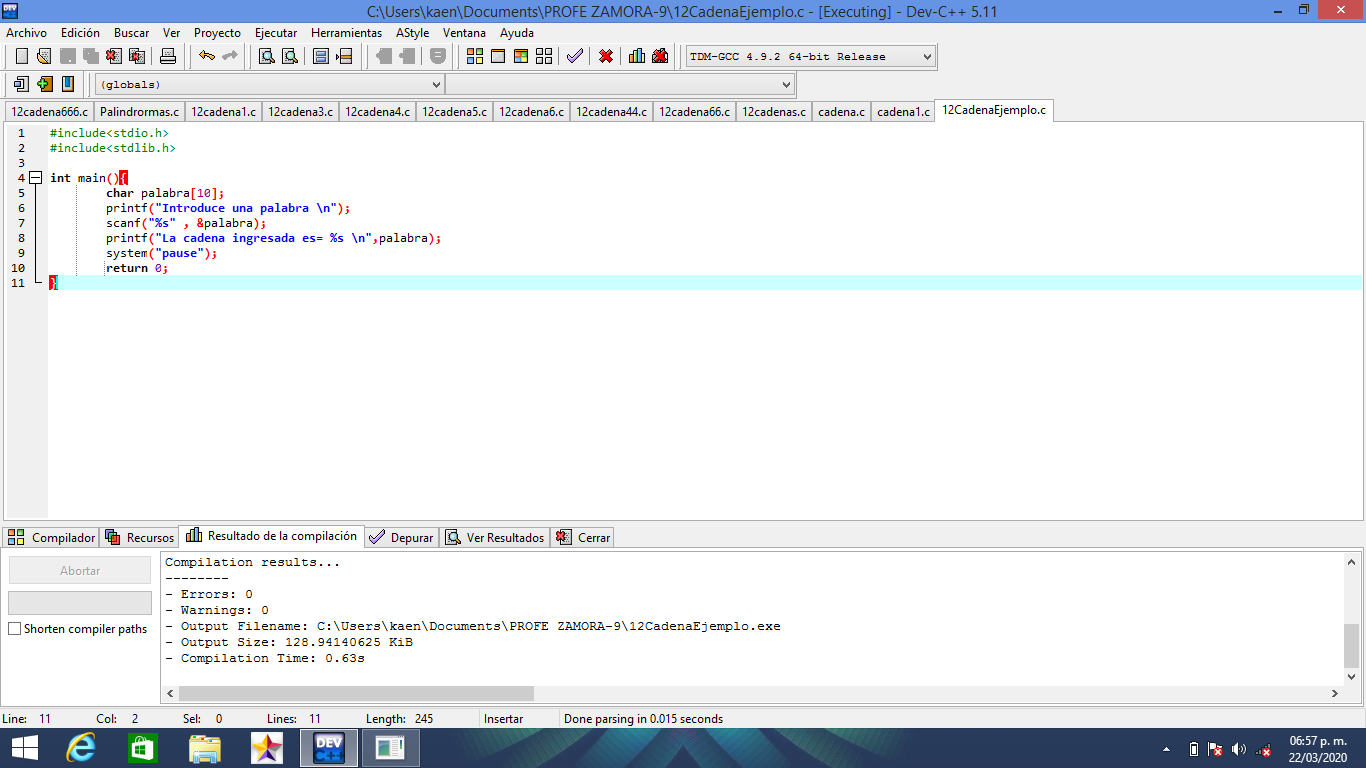
Para poder declarar una cadena es:

char cadena[10];

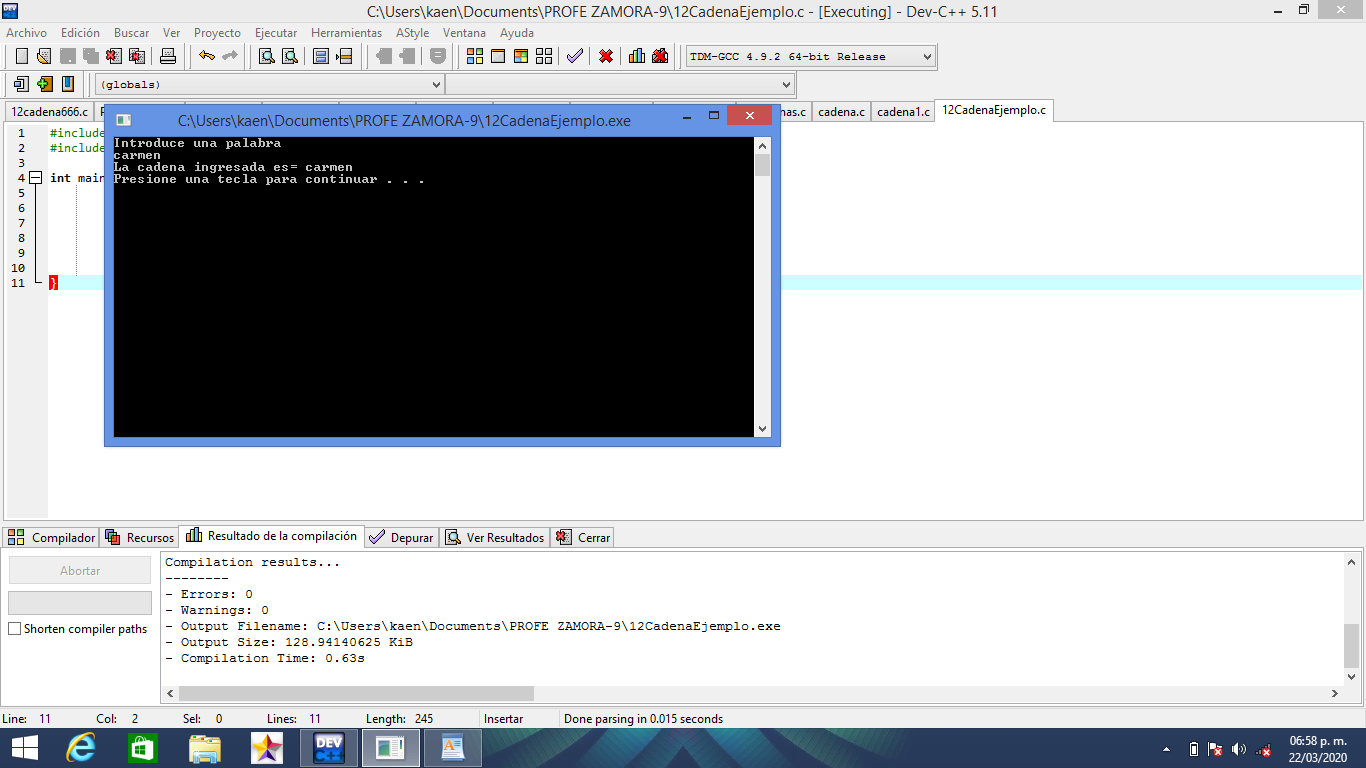
**Ejemplo**

Este es solo un ejemplo de cómo es que se declara una cadena que es de la siguiente manera: char palabra[10]; que es char que indica que va hacer una cadena, el nombre de la cadena, entre corchetes el tamaño de la cadena.

Posteriormente vamos a pedirle al usuario que ingrese una palabra con ayuda del printf, después con ayuda del scanf que este sirve para pedir datos desde teclado el usuario deberá ingresar una palabra, por ultimo vamos a imprimir que palabra ingreso el usuario.



En la pantalla de abajo se muestra el mensaje cuando te pide que ingreses una palabra, la palabra ingresada fue carmen y en la tercera línea nos muestra el mensaje que cadena fue ingresada.



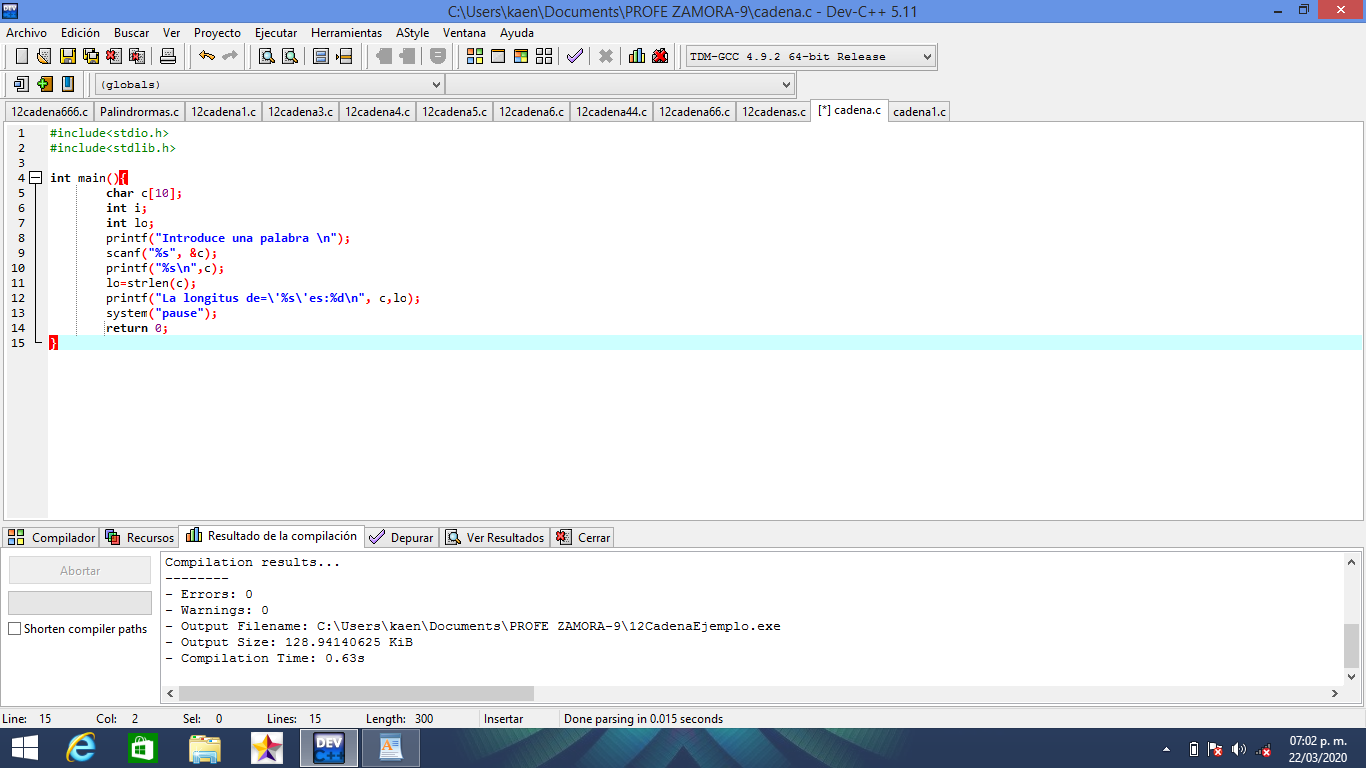
**Ejercicio 2**

El ejercicio consta de ingresar una palabra por teclado y ver el tamaño de la cadena, para eso vamos a ocupar la función que es strlen que esta devuelve la longitud de la cadena sin tomar él cuenta el carácter final de la cadena. Su sintaxis es la siguiente:

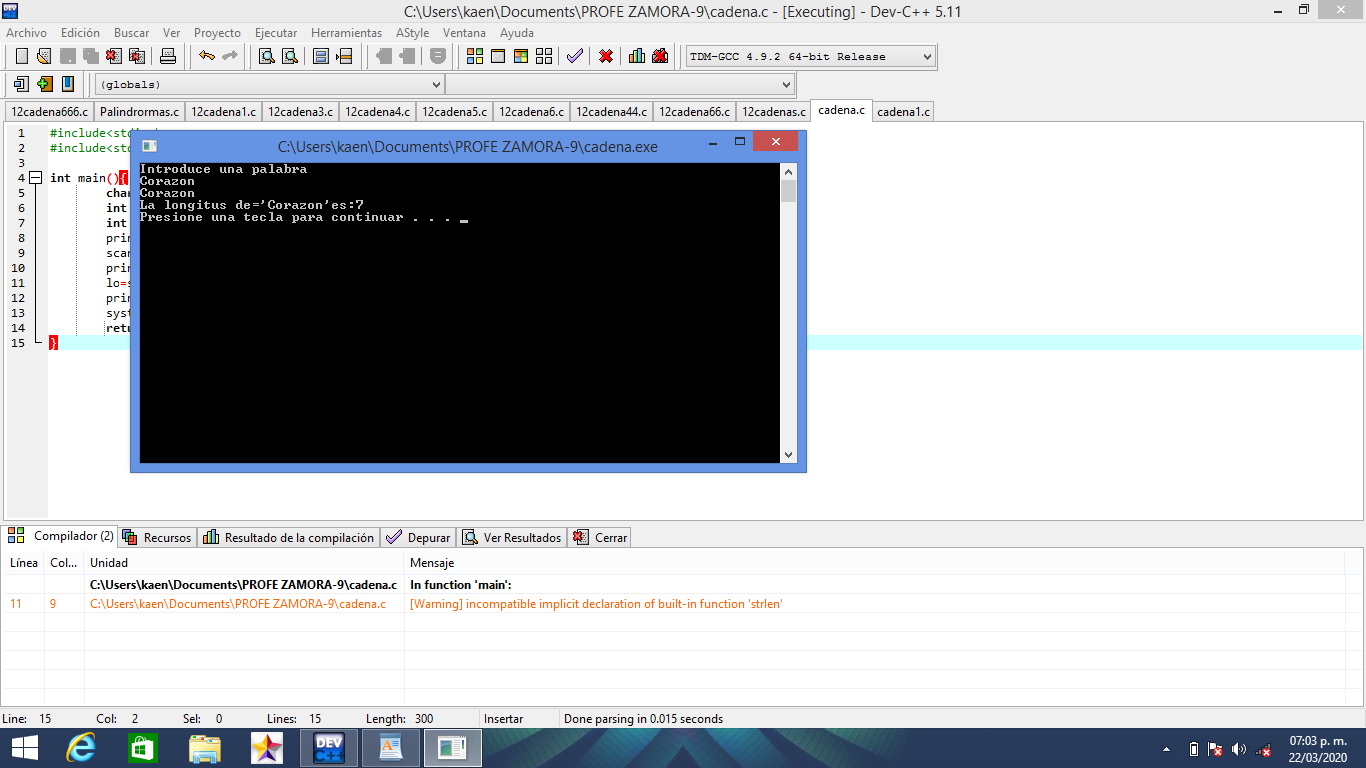
strlen<cadena>

En la pantalla de abajo se muestra la declaración de mi cadena que se llama c y le di un tamaño de 10, tengo dos variables de tipo entero, pero una no la ocupe.

Posteriormente con ayuda de un printf coloque un mensaje que indique que ingrese una palabra, con el scanf pido la palabra por teclado, imprimo la palabra ingresada, en la variable lo voy a guardar el tamaño de la cadena por eso igualo a esa cadena, coloco strlen entre paréntesis la palabra que ingreso el usuario este nos devolverá el tamaño, por ultimo imprimí la cadena y su longitud.



Guardo mi programa y ejecuto, primero manda el mensaje que dice que ingreses una palabra después se tendrá que ingresar la palabra en este caso ingrese corazón y se vuelve a imprimir corazón porque mande a imprimir lo que ingrese, en la cuarta línea indica la palabra que fue corazón y dice que la longitud es 7 y efectivamente esta tiene 7 letras. Como vemos el programa si realiza lo que queríamos.

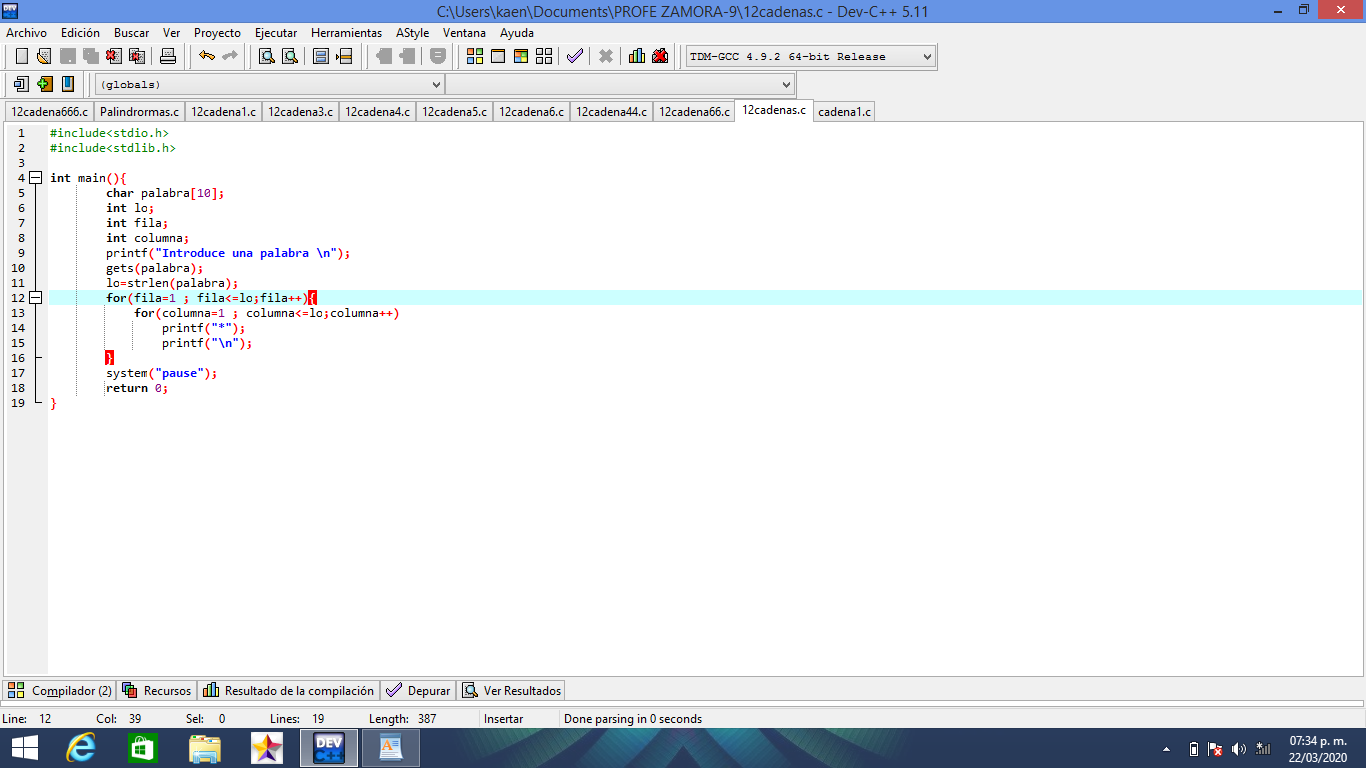


**Ejercicio 3**

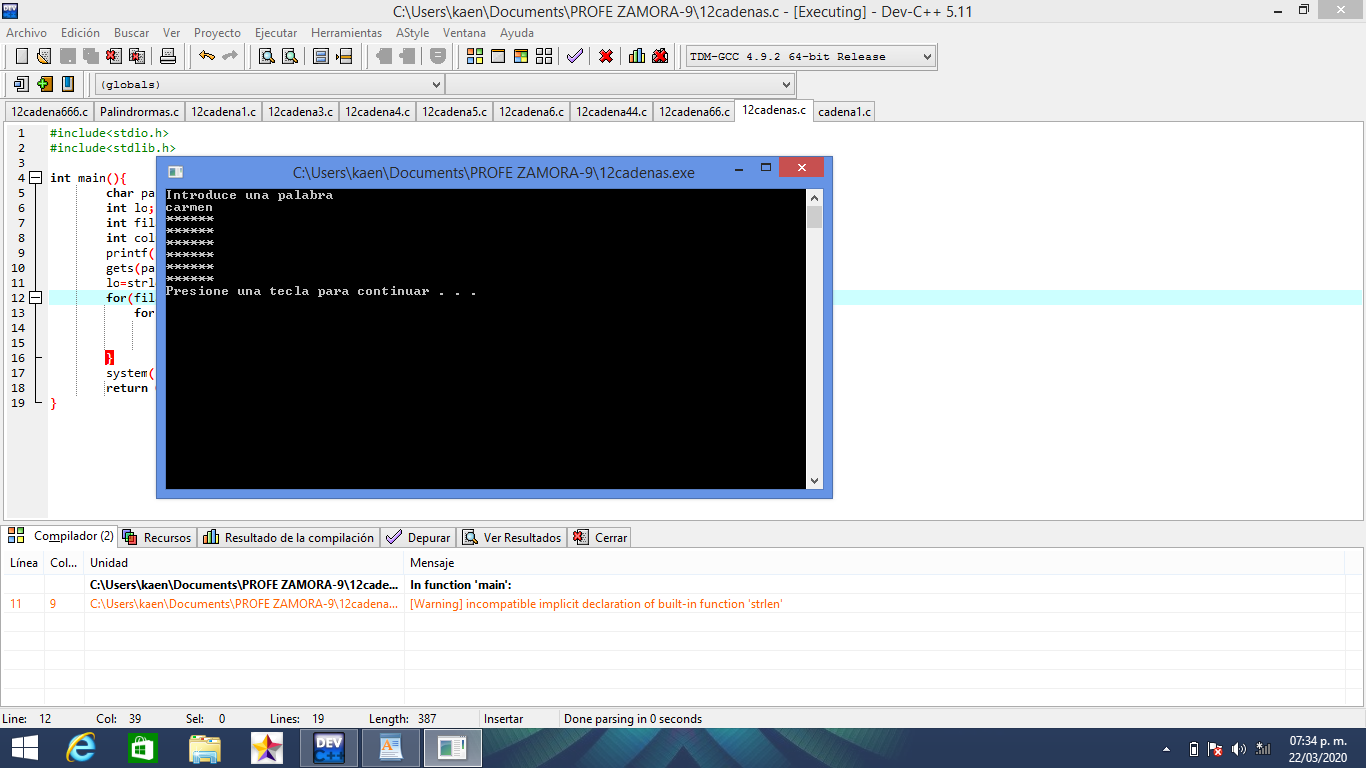
Este ejercicio consta de ingresar una palabra por teclado, dada la palabra tendrá que darte un cuadro de asteriscos de N\*N, para eso sacaremos la longitud de la palabra para poder sacarlo.

Declaro mi cadena que se llama palabra, tengo tres variables de tipo entero que una me ayudara a guardar el tamaño de mi cadena, las otras dos son para ver las filas y columnas que tendrá el cuadro.

Después le pedí al usuario por medio de un mensaje que ingresará una palabra, con ayuda del gets que este también nos ayuda a recibir palabras por teclado pedí la palabra ya que hice eso con ayuda del strlen voy a obtener el tamaño de la cadena, ya teniendo eso con ayuda de un for anidado voy a poder imprimir las filas y las columnas de mi cuadrado que será N\*N como se muestra abajo.



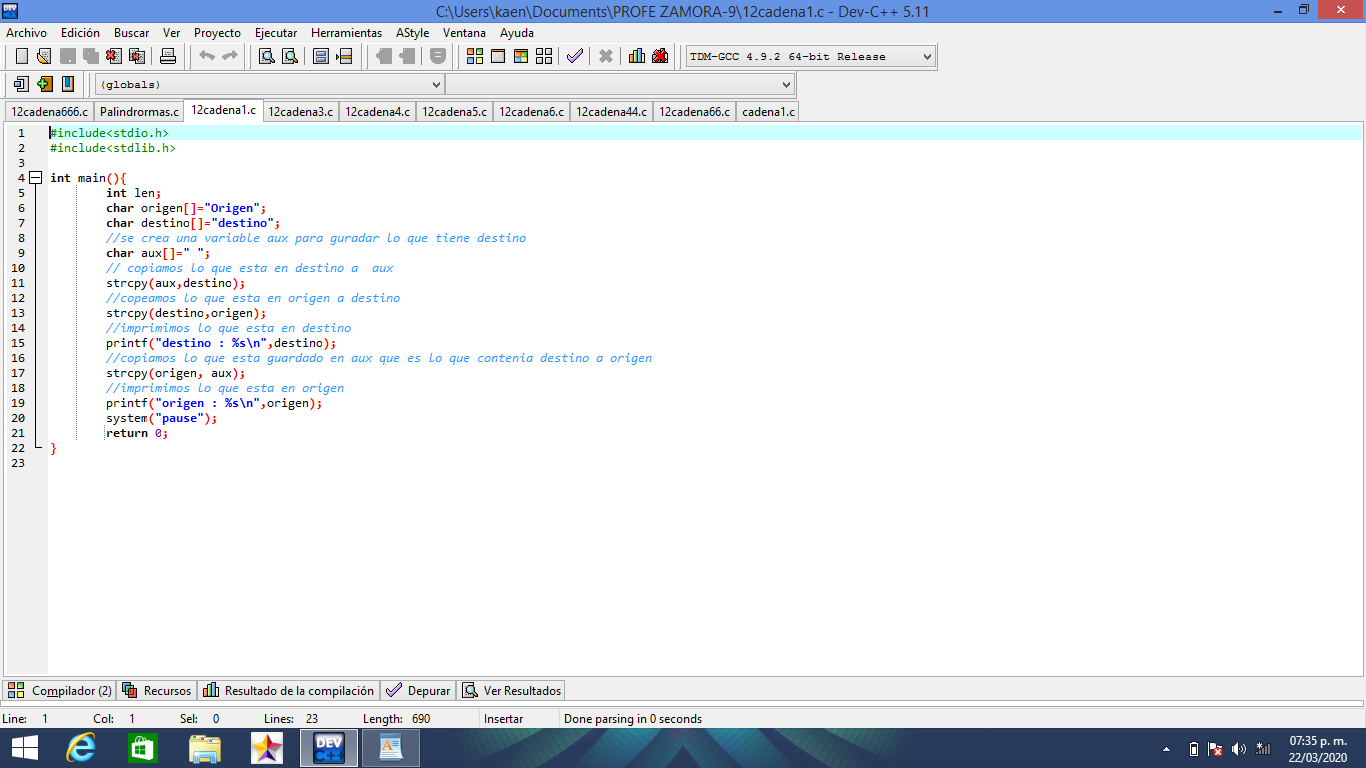
En la parte de abajo se muestra cuando se manda el mensaje para que el usuario ingrese una palabra en este caso ingrese mi nombre Carmen que consta de 6 palabras así que el cuadro será de 6 \*6 que son 6 columnas y 6 filas como se muestra en la parte de abajo.



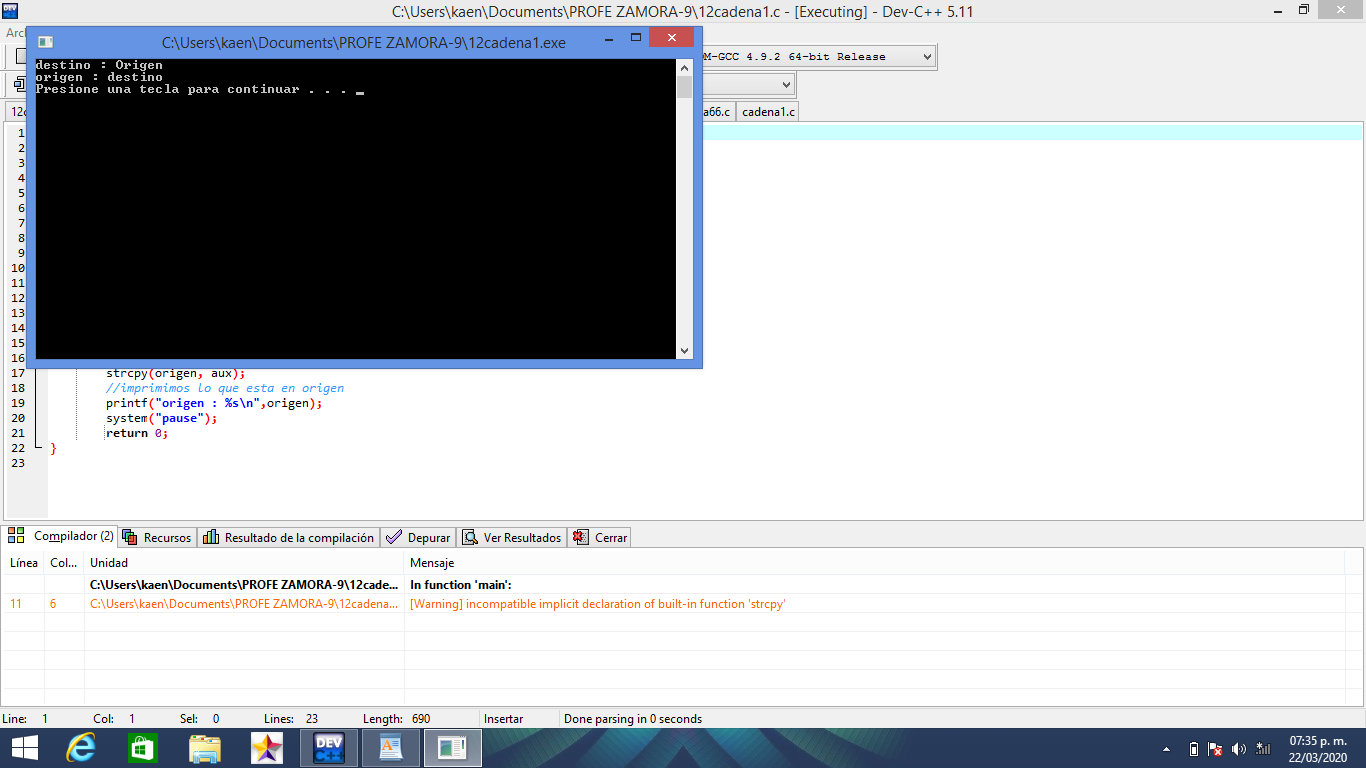
**Ejercicio 4**

El ejercicio consta de declarar 2 cadenas una se llama origen que contenga la palabra origen, otra llamada destino y contenga la palabra destino. vamos a ocupar la función strcpy para copiar el contenido de origen a destino y viceversa.

En la pantalla de abajo se muestra las dos declaraciones de las dos cadenas hay una tercera cadena que nos ayudara a guardar lo que tenga una cadena para que no se pierda a la hora de remplazarse, primero copie lo que hay en destino a auxiliar, después copie lo que había en origen a destino y este guardo la palabra origen y se borró la que tenía, mande a imprimir lo que tenía destino. Posteriormente copie lo que guardo aux a origen e imprimí origen que guardo la palabra destino.



Guarde el programa y lo ejecute y efectivamente destino guardo lo que tenía origen y origen guardo lo que tenía destino.

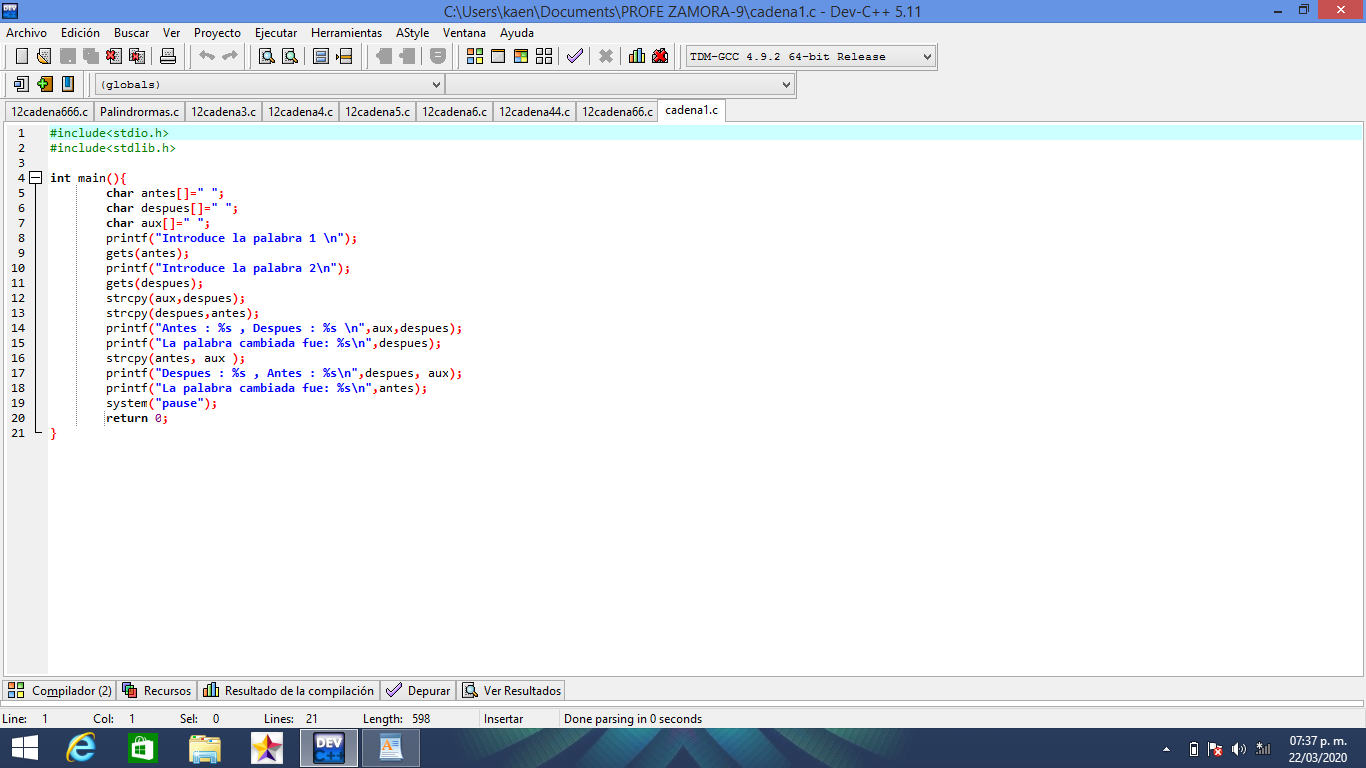


**Ejercicio 5**

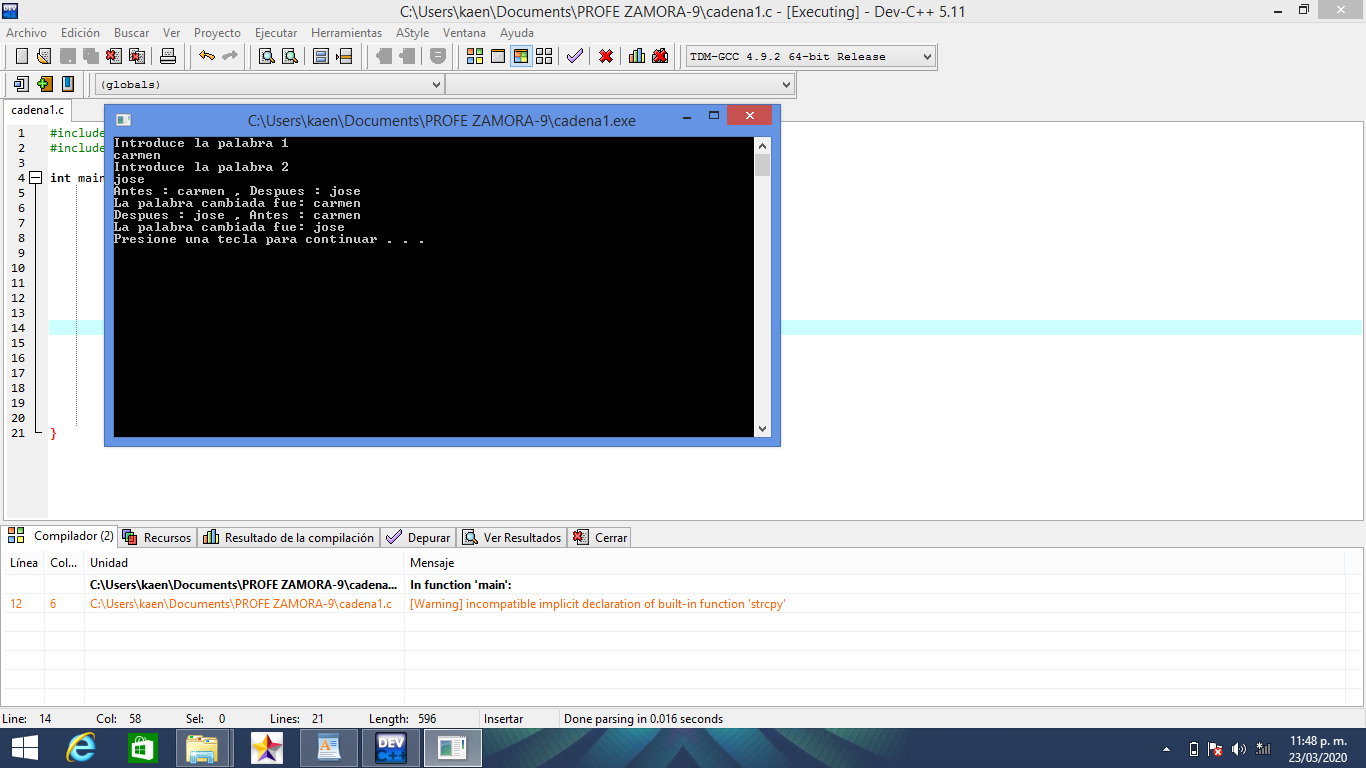
Este ejercicio consta de que le pidamos al usuario que ingrese dos palabras y que estas se intercambien. Haremos uso de la función strcpy para copiar y remplazar lo que ingrese el usuario.

Declare tres cadenas una que guarde la palabra 1 que se guardara en la cadena antes, otra palabra que se guarde en después y el aux para guardar una palabra. Pedí que ingresara el usuario dos palabras, después copie lo que tenía la cadena después a aux, en seguida copie lo que tenía la palabra antes a después y fue remplazada así que mande a imprimir lo que ahora tenía la cadena después y la cadena antes y así pude imprimir lo que ahora tiene después.

Hice lo mismo copie lo que tenía el aux a antes y mande a imprimir lo que tenía antes y después ya que tuviera eso mande a imprimir lo que contenía antes ahora, suena un poco complicado, pero en la ejecución se mira cómo es que funciona.



Guardamos y ejecutamos primero pide ingresar una palabra en este caso ingrese Carmen después pide la segunda palabra e ingrese José recordemos que en aux se guardara lo que contiene después que es la palabra José, copio lo que está en antes a después y se remplaza esta cadena guarda la palabra Carmen como se muestra en la línea 5 antes vale Carmen y después José a la hora de remplazar o copiar vale Carmen que es lo que se guardó en antes, posteriormente en la cadena antes copiamos lo que se guardó en el aux que es la palabra José y efectivamente si da el resultado que queríamos.



**Ejercicio 6**

En este ejercicio hay que utilizar la función de strcat que nos sirve para poder concatenar dos cadenas, el ejercicio nos dice que el usuario ingresara dos palabras y concatenemos N veces la segunda palabra a la primera palabra, donde N es la longitud de la primera palabra,

Ejemplo:

Ingresamos para como primera palabra y como segunda brisas, la longitud de la primera palabra es 4 por que son las letras que la conforman, así que imprimimos la primera palabra y concatenado a esa tendrá que imprimirse 4 veces la palabra brisas por que la palabra tiene una longitud de 4 en la parte de abajo se muestra cómo es que se debe de mostrar.

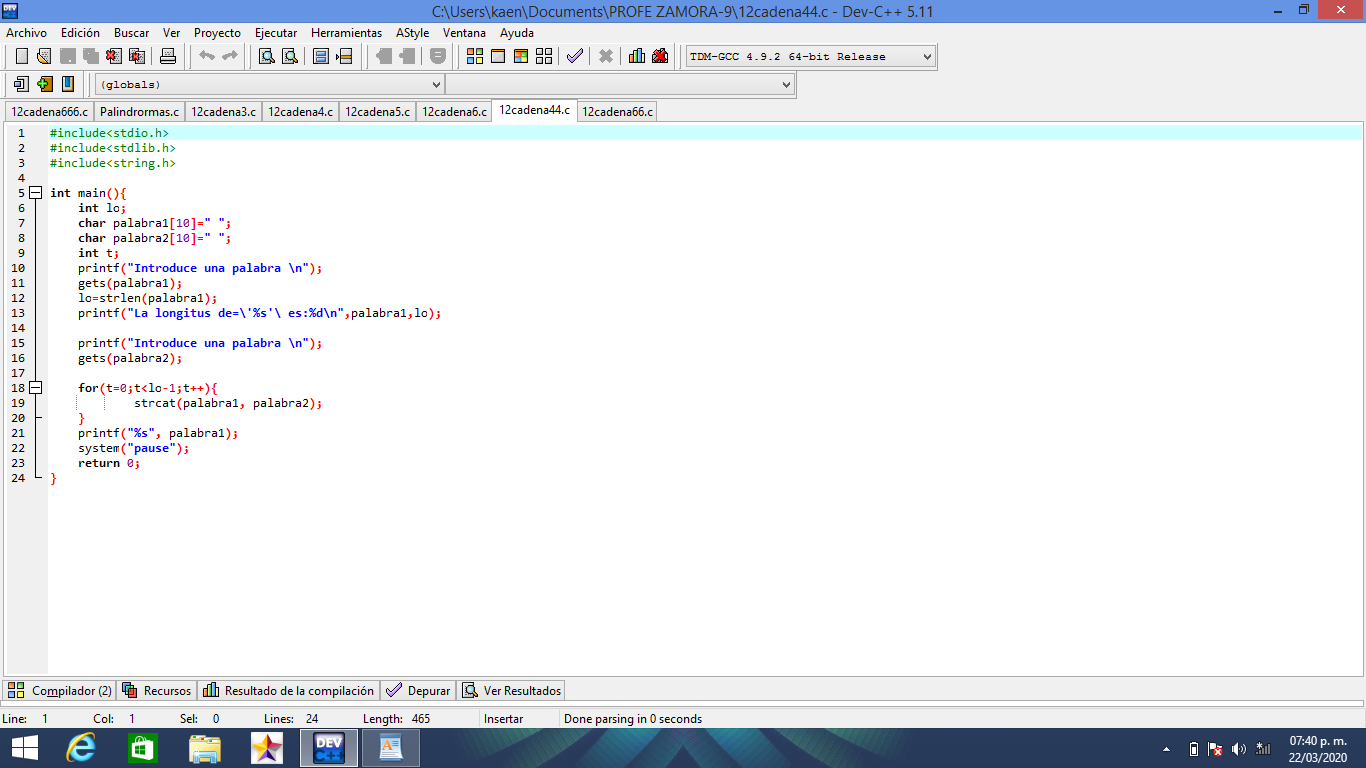
palabra1= para

palabra2=brisas

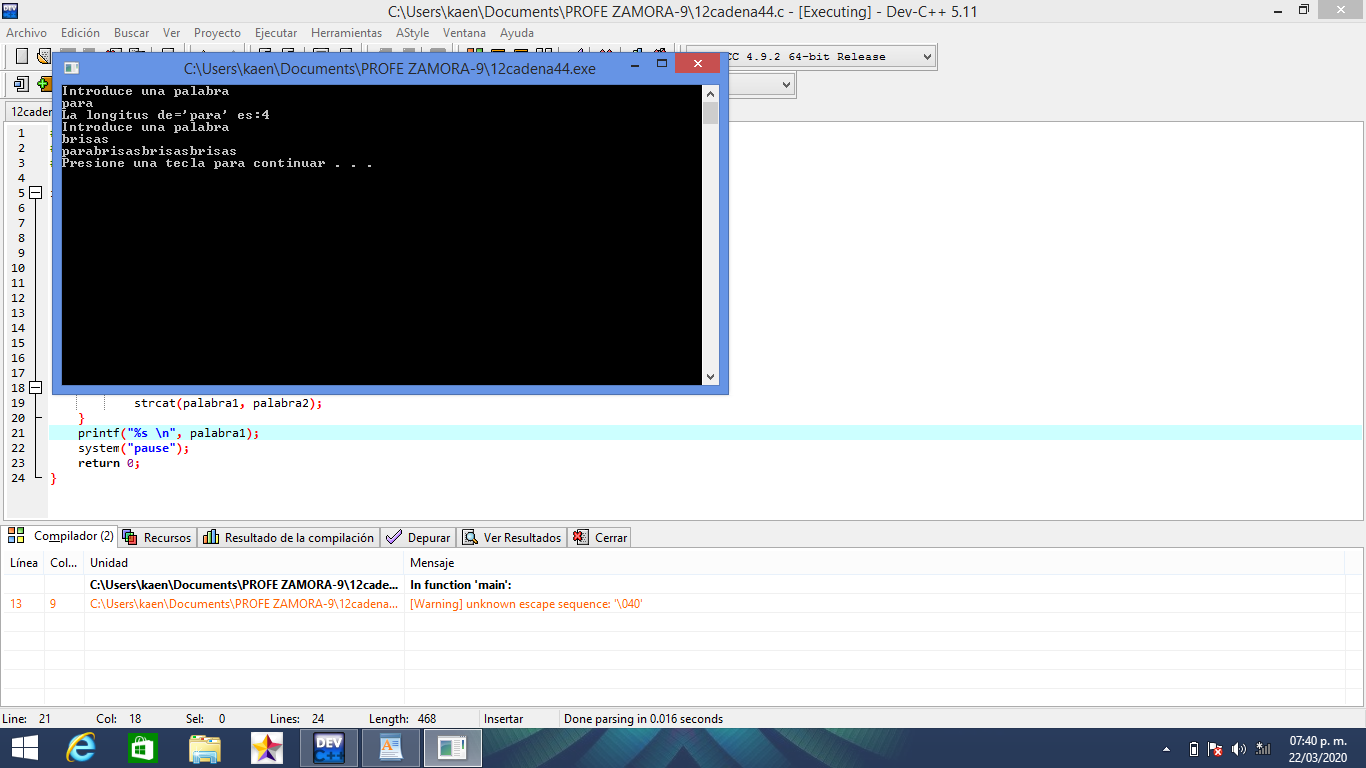
longitud de la primera palabra = 4

parabrisasbrisasbrisasbrisas

ya que entendí como se tenía que hacer el ejercicio pedí al usuario ingresar 2 palabras, posteriormente con la variable lo me ayudara a guardar la longitud de la primera palabra, con ayuda de un for voy a concatenar lo que se me pide que es imprimir la palabra1 y concatenarle la palabra 2 según su longitud.



Guardo y ejecuto ingresa la palabra 1 que es para y la segunda que es brisas y en la línea 6 se imprime lo que queríamos obtener.



**Ejercicio 7**

En este ejercicio vamos a ocupar la función strcmp que nos ayuda a ver si la cadena es más grande más chico o son iguales para esto existen las siguientes condiciones;

si cadena1 = a cadena2 nos devolverá un 0

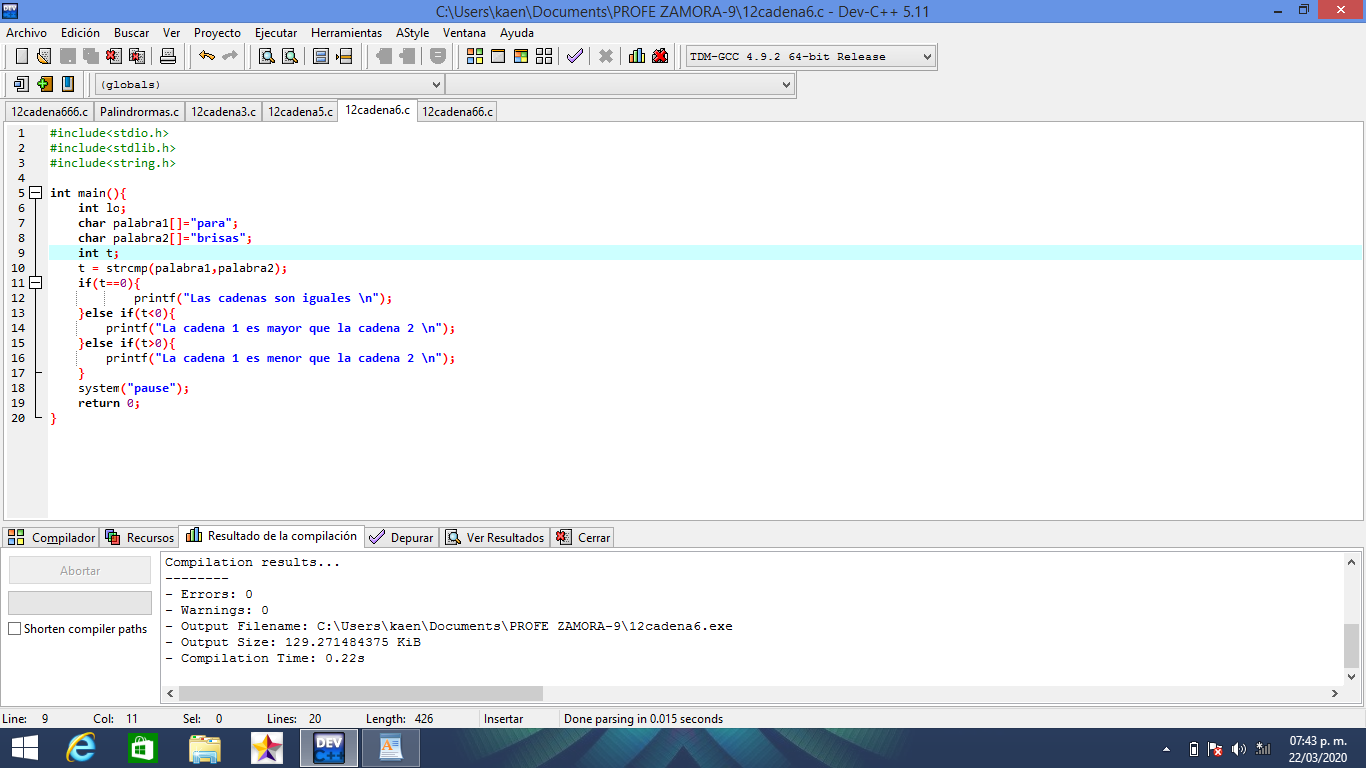
si cadena1 < cadena2 nos devolverá un 1

si cadena1 > cadena2 nos devolverá un -1

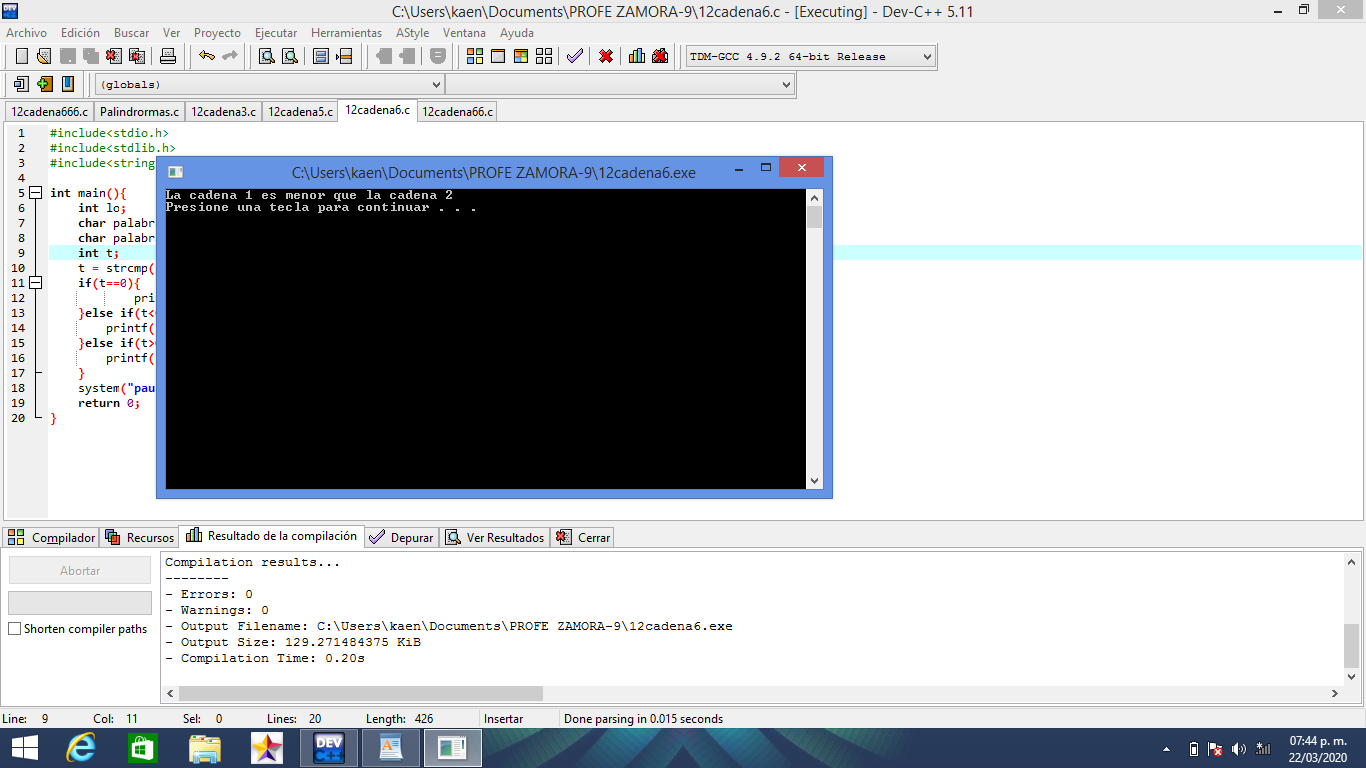
Para ver cómo es que funciona declare 2 cadenas una que me guarde para y otra que me guarde brisas y una variable de tipo entero que esta guardara el número que nos indicara si es mayor o menor o tal vez igual.

coloca la variable t y la igualo con la función strcmp y dentro de ella las dos palabras para ver cuál es mayor o menor o igual y se guardara en t.

Con ayuda de un if anidado voy a obtener el resultado dependiendo de que sea.



Guardamos y ejecutamos y la palabra para < que brisas así que nos muestra que la palabra 1 es menor que la palabra2.



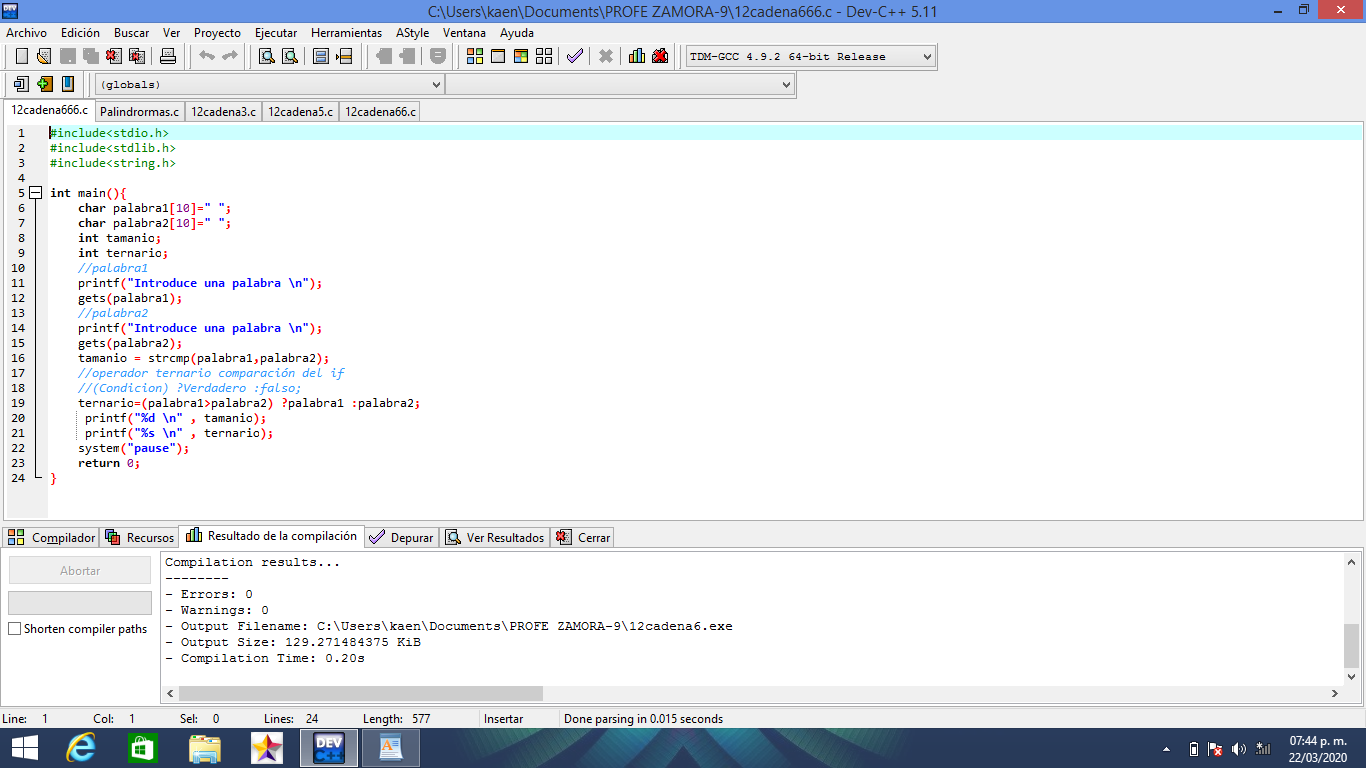
**Ejercicio 8**

Este ejercicio consta de ingresar dos palabras por teclado y que te diga que palabra es más mayor o menor o tal vez son iguales.

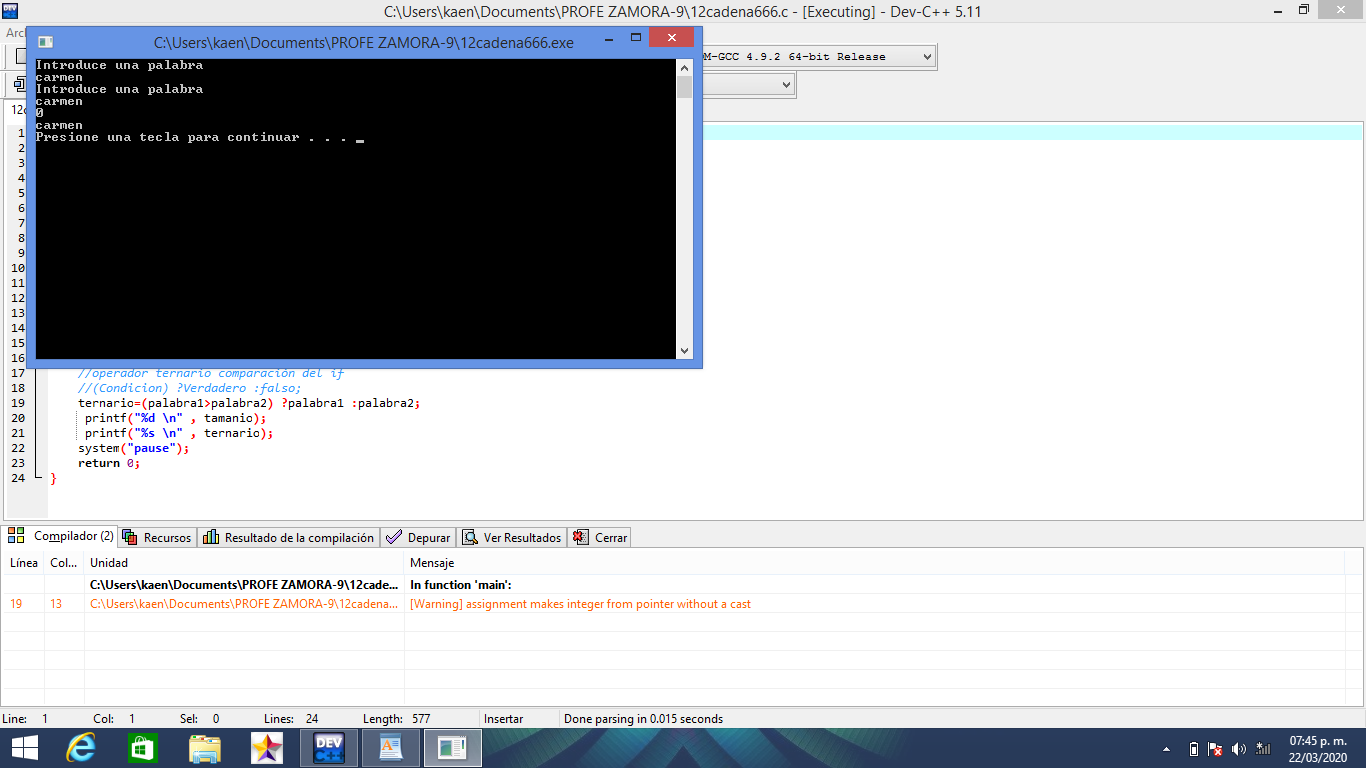
Para esto el profesor nos pidió realizar el ejercicio con el operador ternario que este es como es if pero te reduce código y el más claro.

Su sintaxis es = (condición)? verdadero: falso

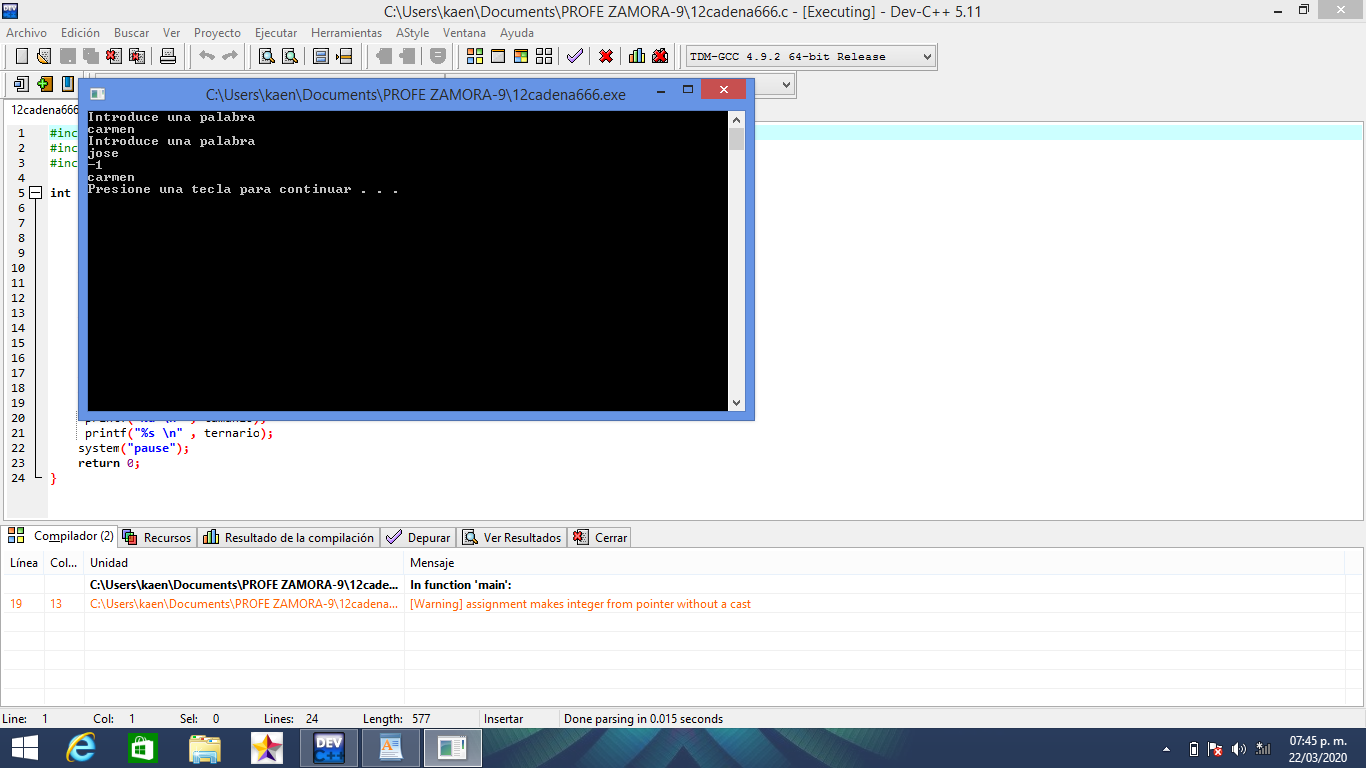
que entre paréntesis va la condición si esta se cumple se imprimirá lo que este en verdadero y si no se imprimirá lo que está en falso. Pedí que el usuario ingresará 2 palabras posteriormente saque con la función strcmp si era mayor o menor o iguales solo pala comparar lo que me dé en el operador ternario, después coloque el operador ternario y en la condición puse si la palabra1>palabra2 si se cumple se imprimirá la palabra1 si no se imprimirá la palabra 2 y si son iguales también se imprimirá la palabra2.



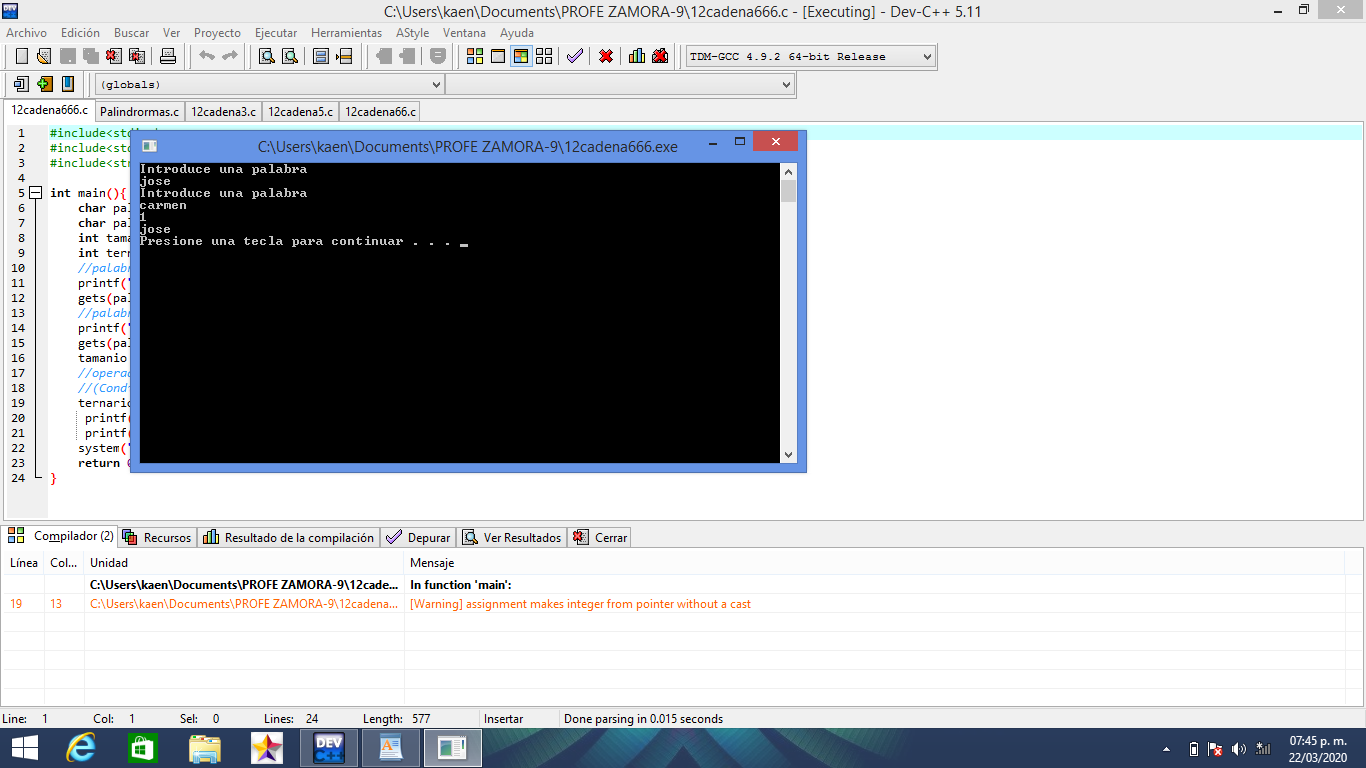
Guardo y ejecuto ingreso la primera palabra que es Carmen y la segunda que también es Carmen así que me devolverá un 0 por que ambas son iguales y la palabra Carmen porque son iguales y se imprime lo que hay en la palabra2.



Después lo volví a ejecutar y coloque Carmen y José que Carmen es mayor que José por eso se imprime -1 y Carmen porque esta es mayor que la segunda.



Por ultimo hice una última prueba ingrese José y luego Carmen José es menor que Carmen así que nos dará 1 y se imprimirá José por que la condición se cumple y se imprime lo que está en falso.



**Ejercicio 9**

Este ejercicio consta de hace un programa que te diga si las palabras son palíndromas o no esto se es que al decir una palabra al revés se diga igual, por ejemplo:

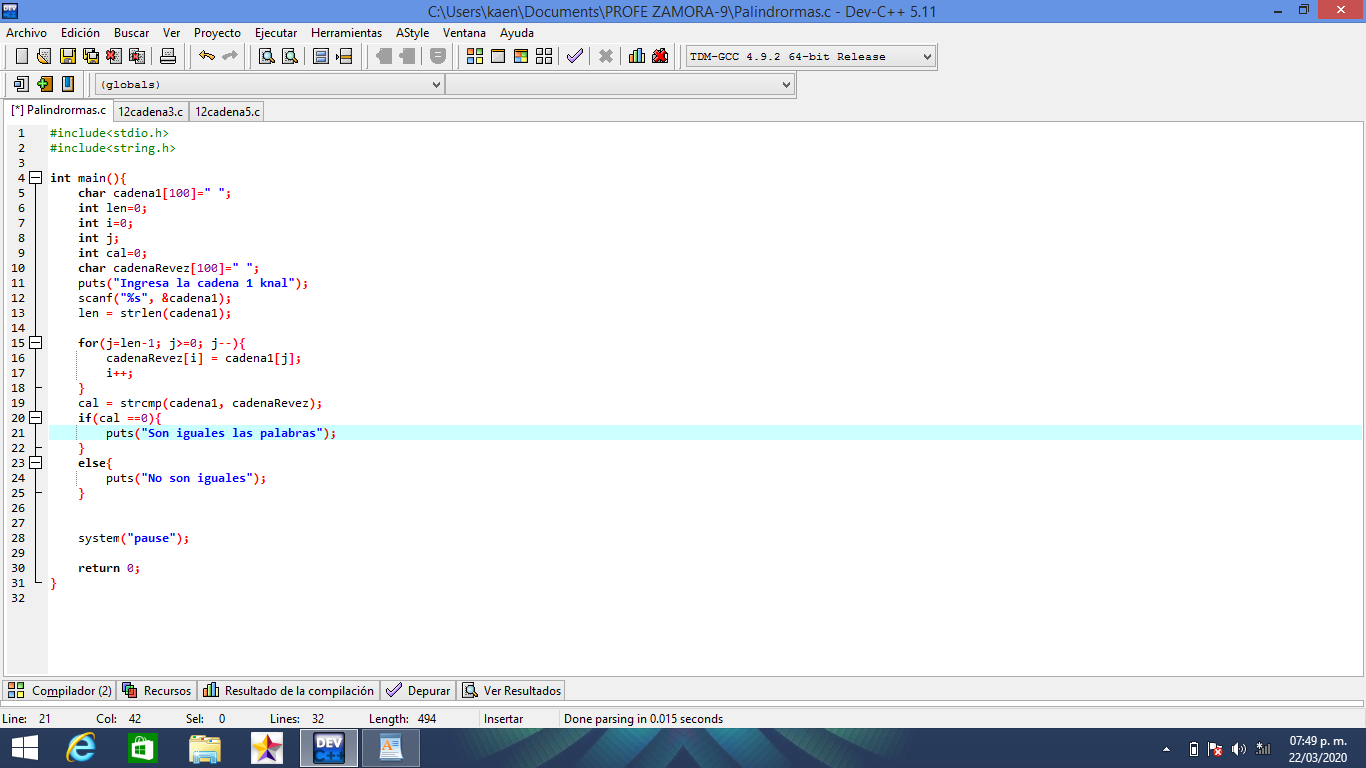
Ana

Oso

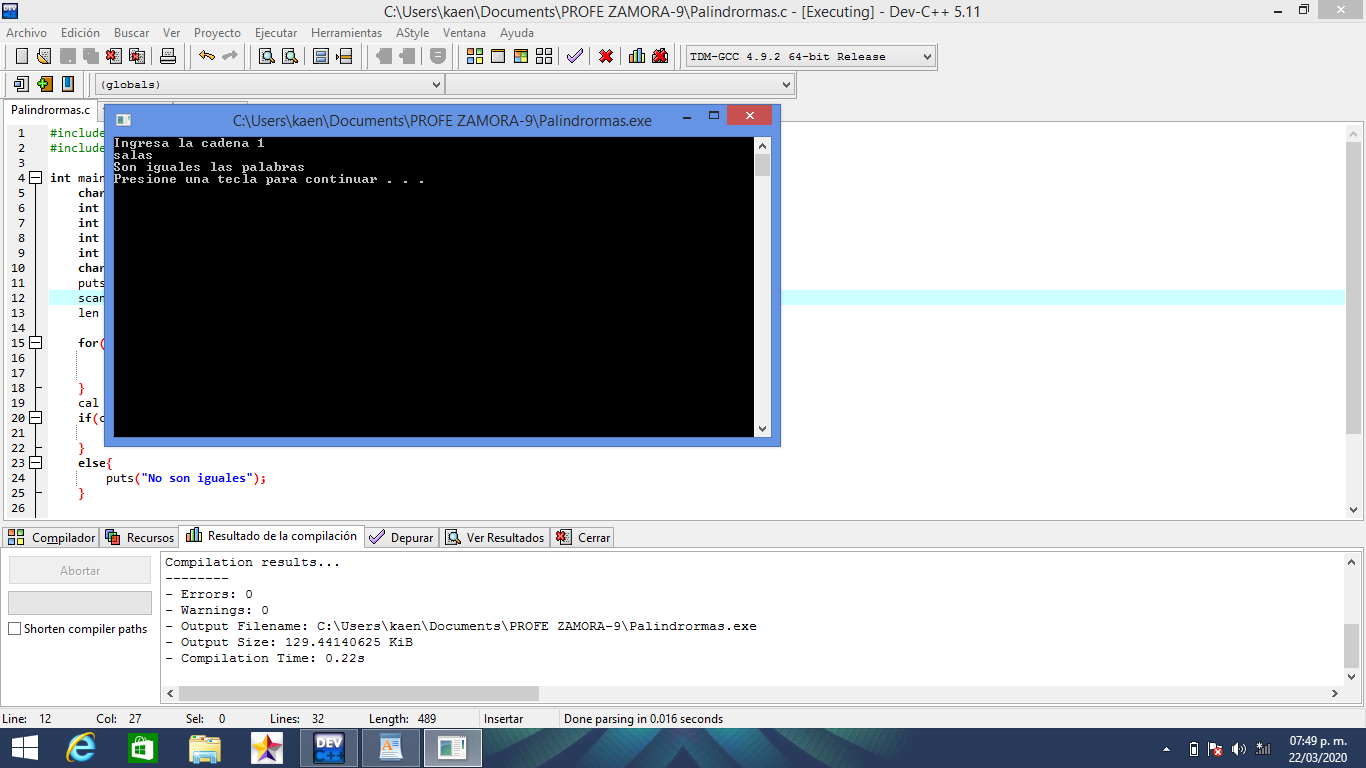
Salas

Para poder realizar este ejercicio declare mi cadena llamada cadena1, ingrese 4 variables de tipo entero que me ayudaran a poder saber y evaluar por ultimo declare otra cadena que me ayudara a guardar la cadena al revés.

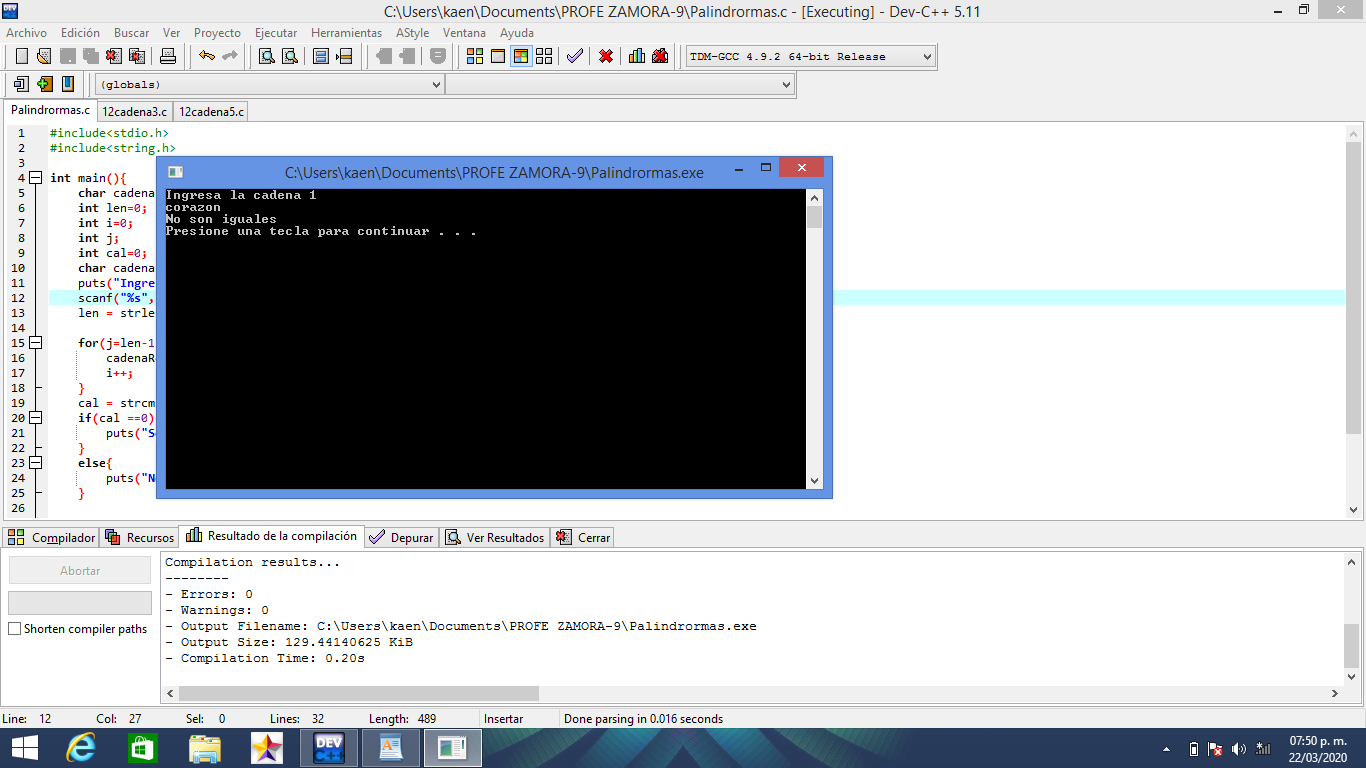
Pedí al usuario ingresar una palabra, después saqué su tamaño y con ayuda de un for anidado pude saber si las palabras son iguales o no.



Guarde y ejecute ingrese la palabra salas y mostro que las palabras son iguales por que esta al revés se dice igual.



Hice otra prueba e ingresé corazón y nos manda un mensaje de que no son iguales por que corazon al revés es nozaroc y no dice lo mismo



**13.Funciones**

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).

1. Facilita el diseño descendente.

2. Los procedimientos dentro de ellas se pueden ejecutar varias veces.

3. Facilita la división de tareas.

4. Se pueden probar individualmente

5. Con funciones apropiadamente diseñadas, es posible ignorar como se realiza una tarea, sabiendo qué es lo que hacen

Ventajas

1. Funciones diseñadas para realizar operaciones a partir de sus argumentos y de volver un valor basado en sus cálculos.

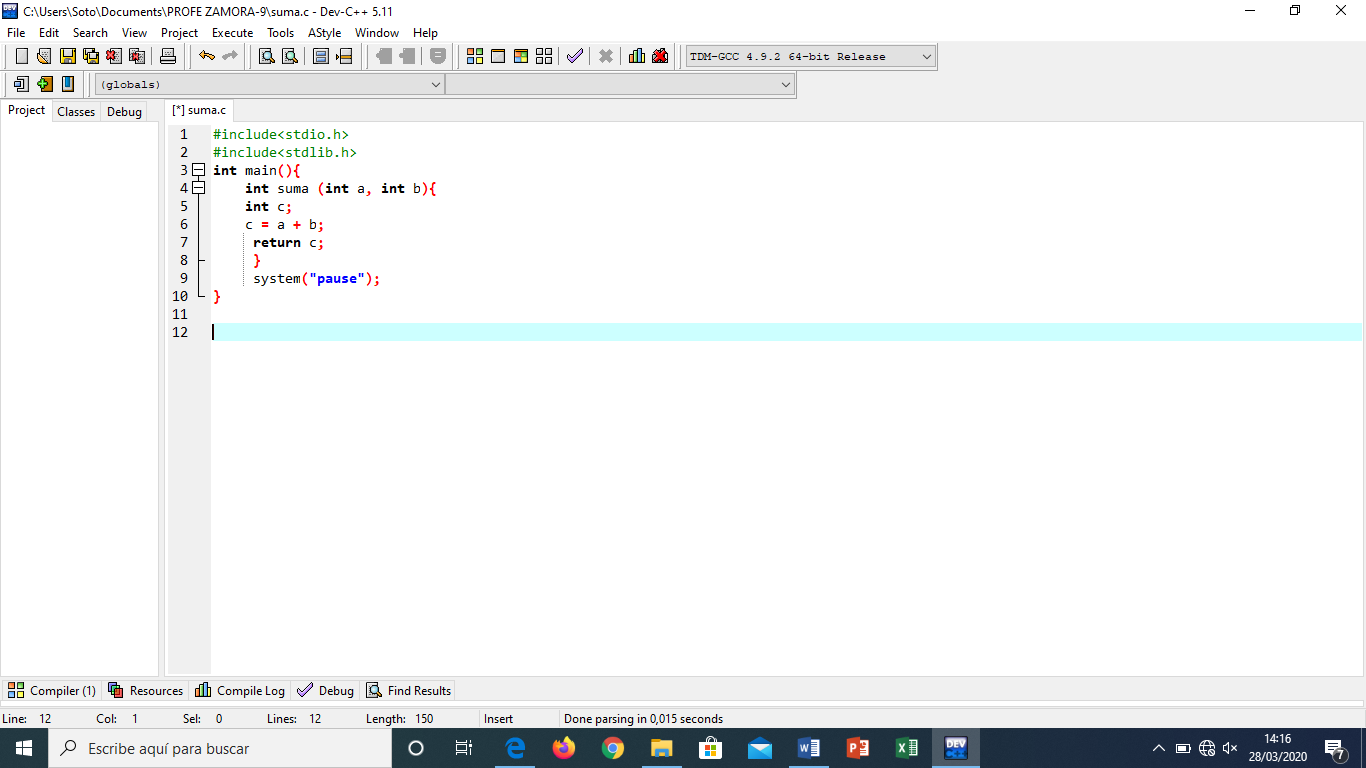
Modo de uso

2. Funciones que no reciben argumentos, realizan un proceso y devuelven un valor .

3. Funciones que no tienen argumentos ni valor de retorno explícito, operan sobre el entorno de variables globales o atributos del sistema operativo.

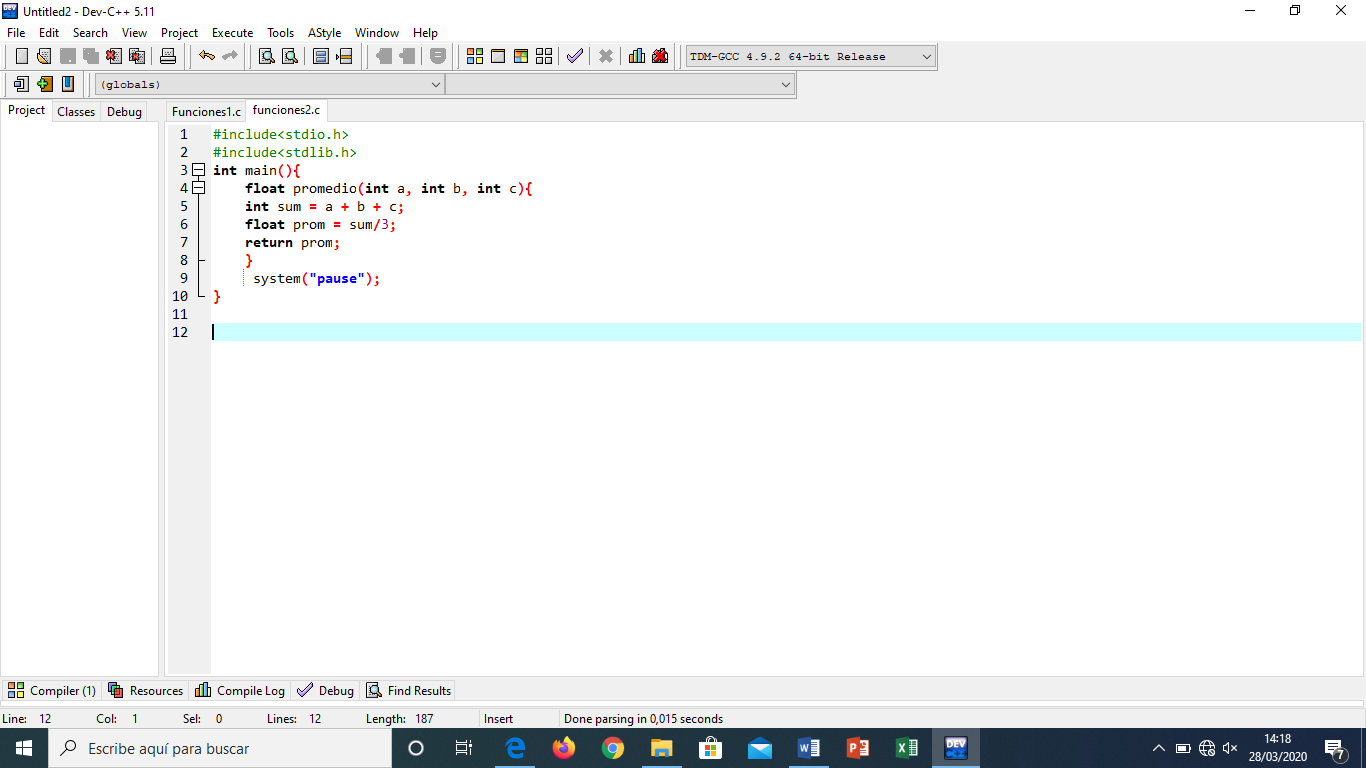
**Ejercicio 1**

En la imagen de abajo se muestra un pequeño programa que nos indica cómo es que se usan las funciones, para esto declaramos en método main dentro de él se creó otro método que es de tipo entero, llamado suma y tiene dos parámetros de tipo entero, dentro de las llaves esta lo que va a realizar y es sumar a + b y el resultado guardarse en c que es de tipo entero posteriormente nos retornara c.



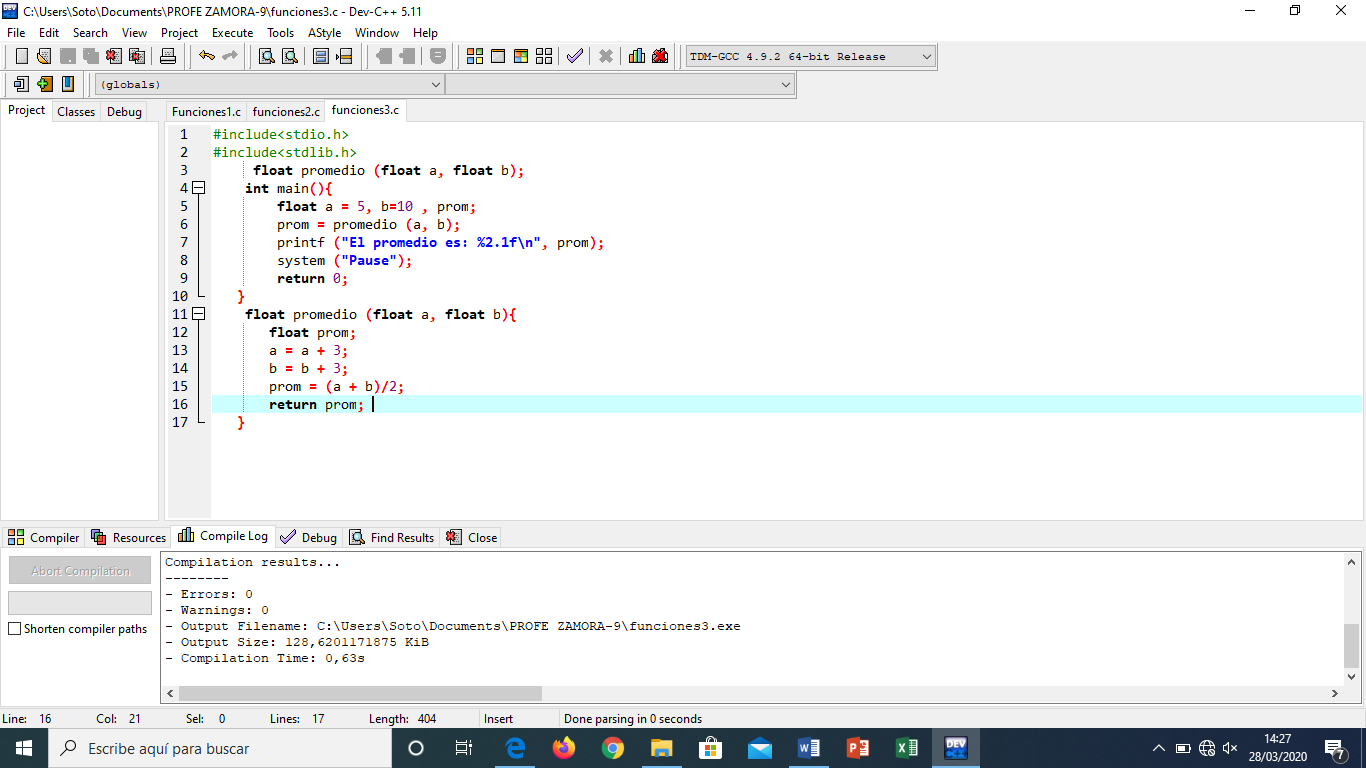
**Ejercicio 2**

En el ejercicio de abajo se muestra como se saca la suma y el promedio usando funciones para esto declaramos el metido main dentro de el un método de tipo flotante con el nombre de promedio y con tres parámetros que, recibida de tipo entero, ya que ara la suma de los tres y posteriormente sacara el promedio de la suma entre tres y nos regresara el promedio.



**Ejercicio 3**

Se declara una variable de tipo flotante con parámetros.



Tenemos el método que nos ayudara a realizar las operaciones de tipo flotante llamado promedio con vos parámetros a y b de tipo entero, una variable de tipo flotante que guardara el promedio. Empezamos a=5+3; y b= 10+3; el prom= (8+13)/2;

El promedio será igual a 10.5;

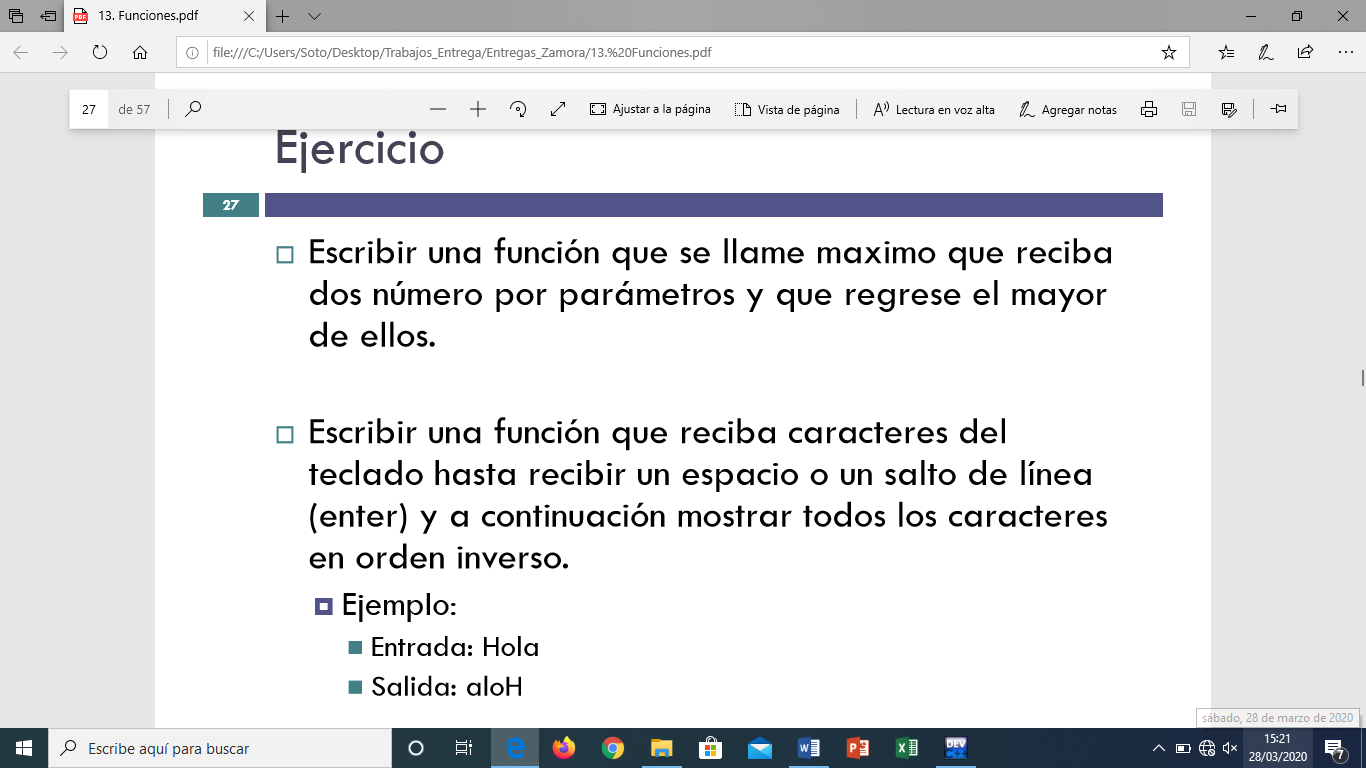
Dentro del método main colocamos 3 variables de tipo flotante a = 5, b=10 y prom.

Prom que será igual a la variable donde guardamos los argumentos y se imprimirá 10.5 el promedio total.

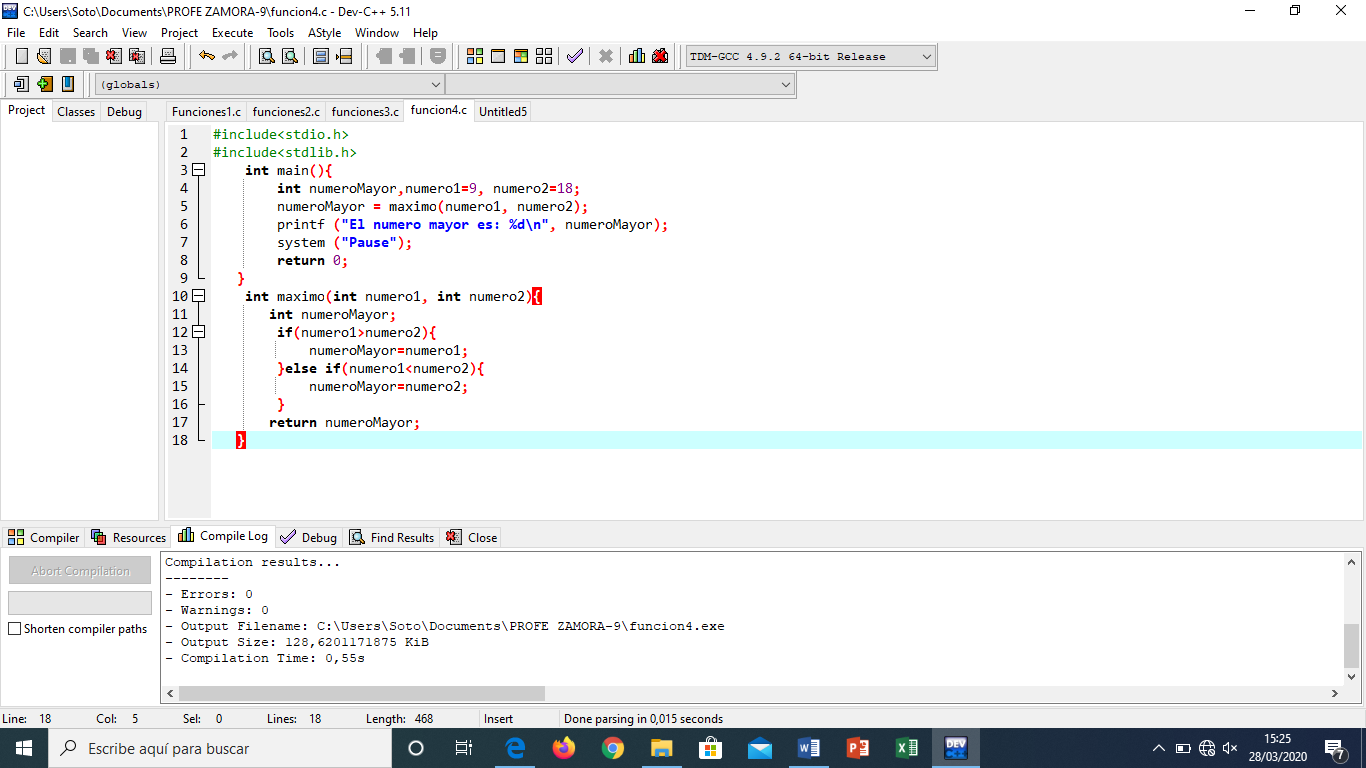
En la pantalla de abajo se muestra el resultado obtenido. efectivamente nos da 10.5.



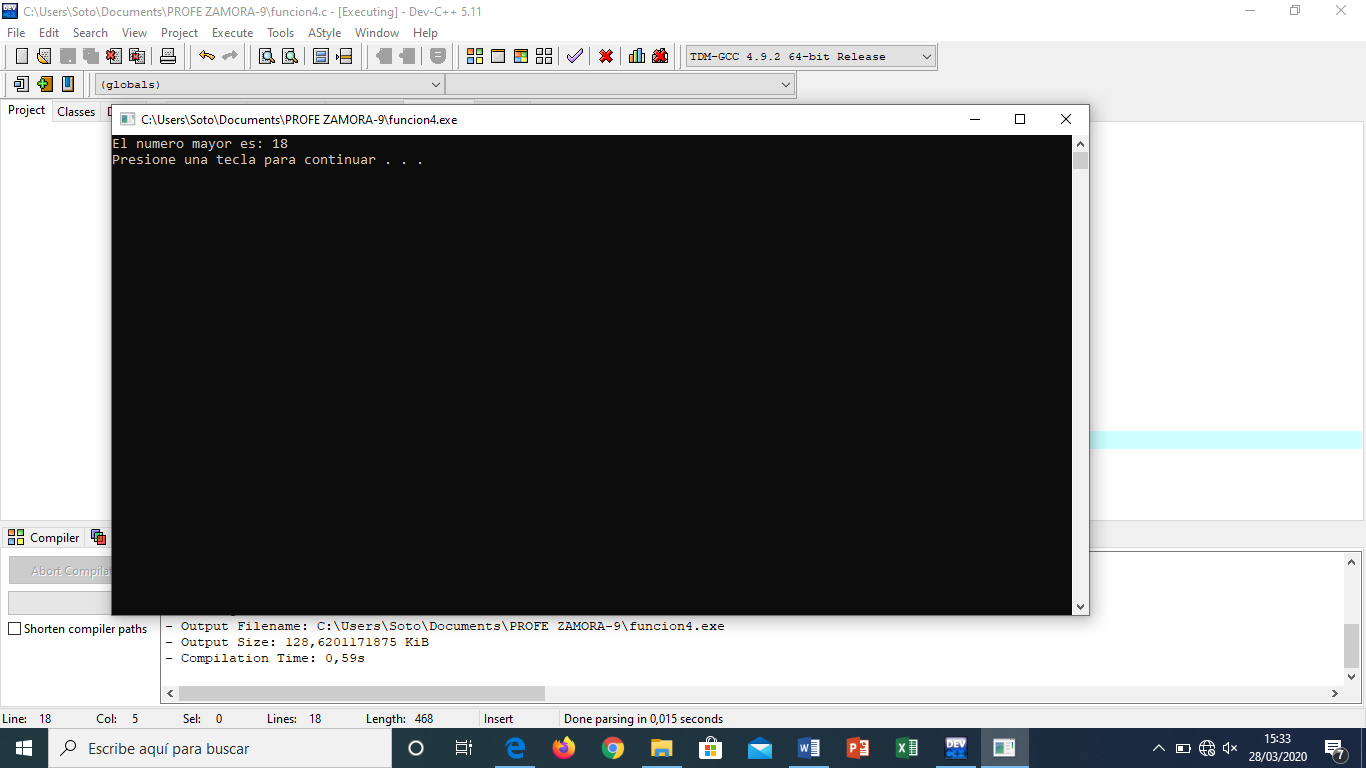
**Ejercicio 4**



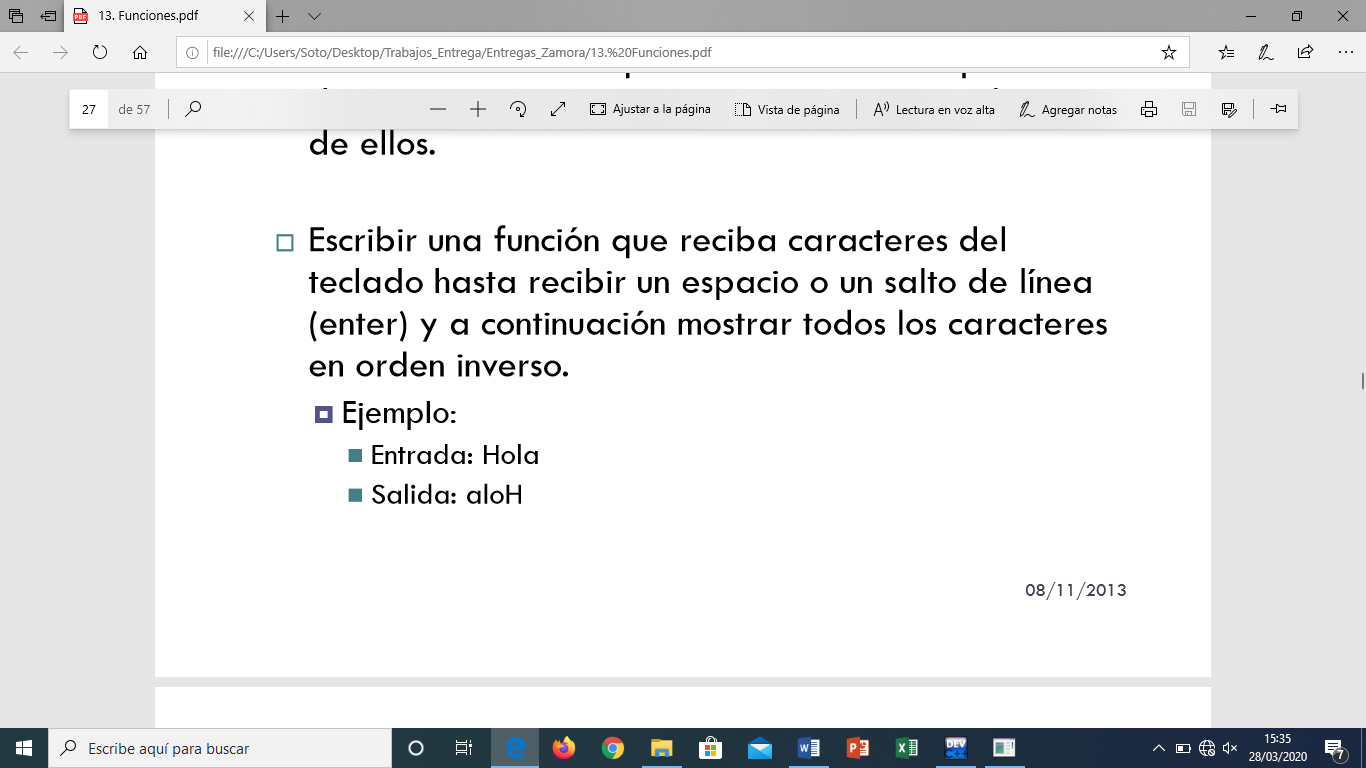
Este ejercicio lo resolví con ayuda del anterior porque es parecido, coloque primero mi método principal, coloque tres variables de tipo entero una que guarde el número mayor, otras dos que guardaran el número 1 y 2 para ver cuál es el mayor, después esta la variable numeroMayor y va hacer igual a lo que nos retorne el método máximo, así que nos vamos a l método es de tipo entero tiene 2 parámetros de tipo entero que es numero1=9 y numero2=18 posteriormente esta la variable que nos guardara el número mayor y que se retornara, con ayuda de un if anidado voy a comparar cuál de los numero es mayor la primer condición es si numero1>numero2 se guardara en la variable numeroMayor el número y si no se cumple se ira a la segunda condición que si numero1 es menor que numero2 se guardara en la variable numeroMayor = numero2. y se retornará la variable numeroMayor que esta se imprimirá en el método principal junto con el mensaje mostrando que número es mayor.



En la pantalla de abajo se muestra que el número mayor es el 18 que estaba guardado en la variable numero2.



**Ejercicio 5**

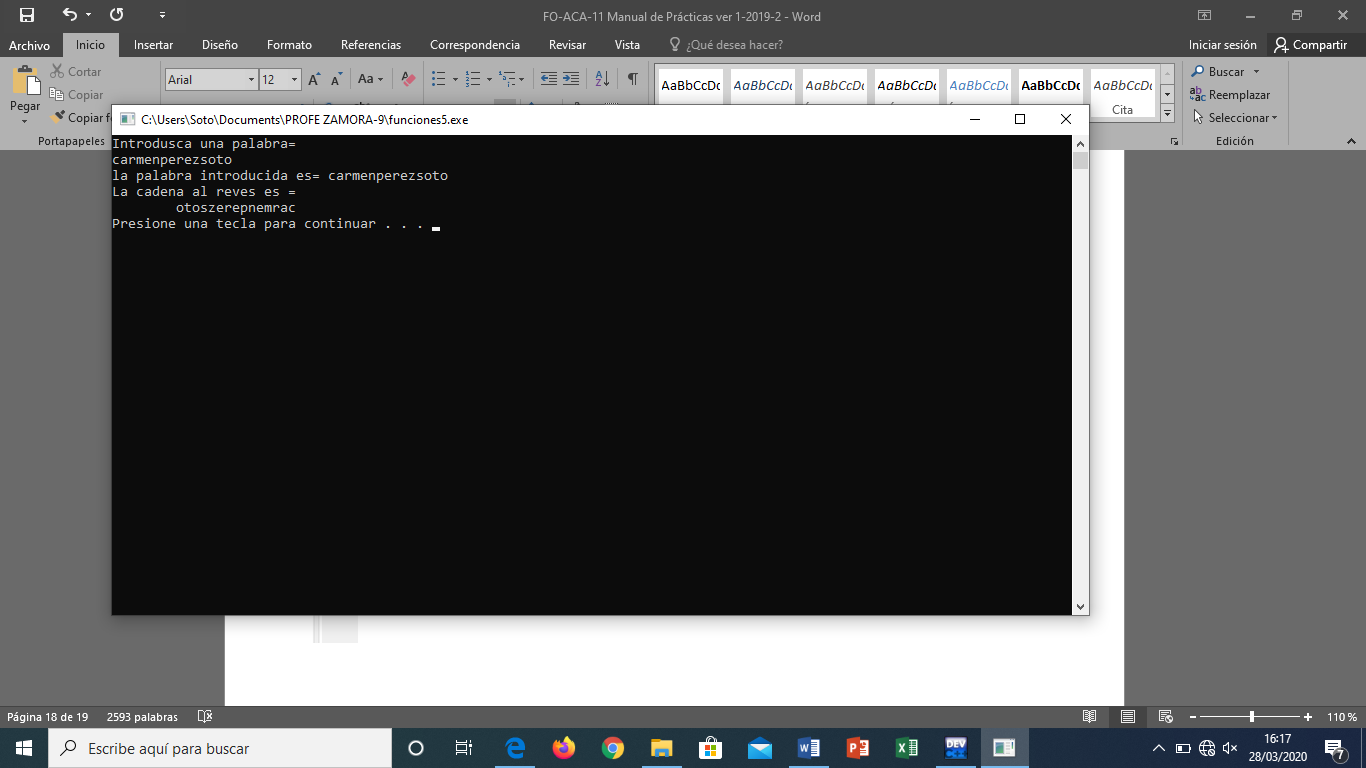


Este ejercicio se me complico mucho por lo de los caracteres, no lo pude resolver por medio de una función porque me era difícil retornar su valor y aparte no supe cómo es que una variable de tipo char guarda lo que retorna el método.

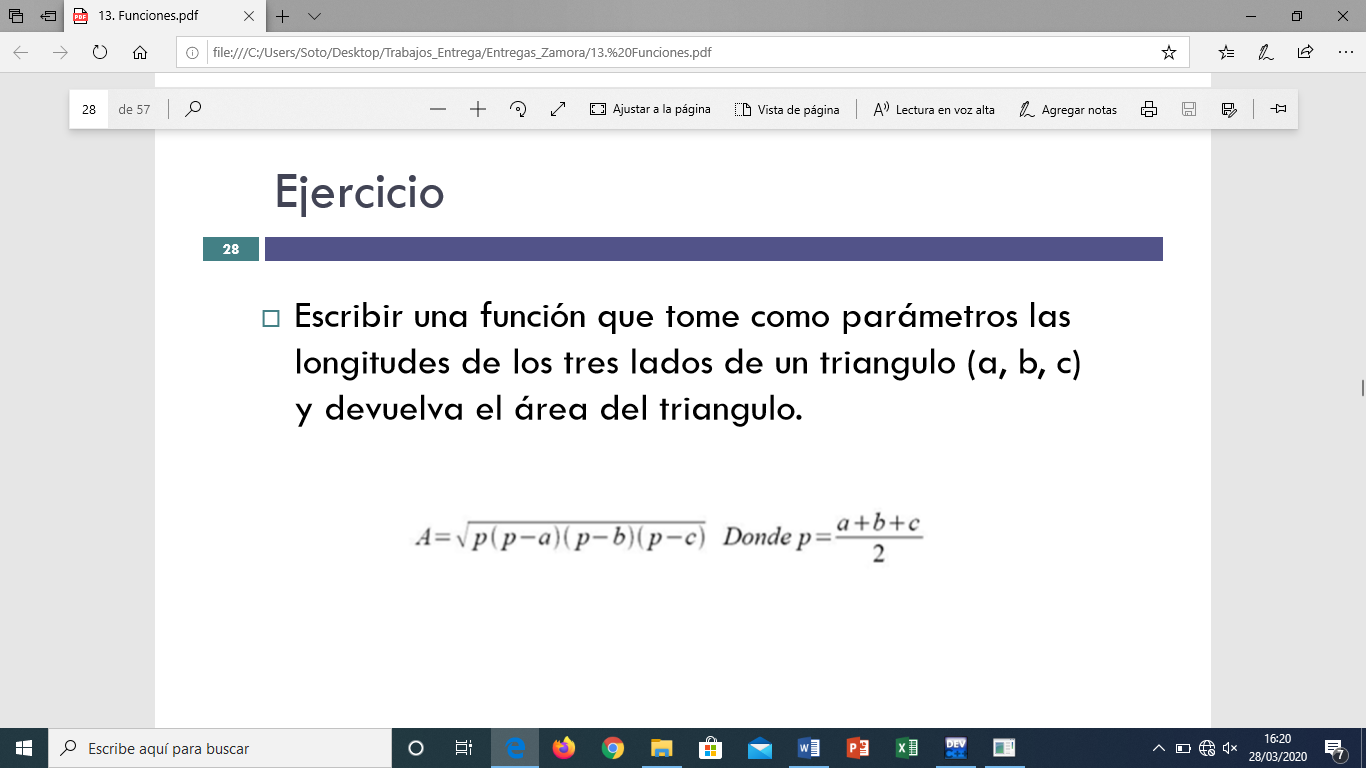
Asi que lo hice solo pidiendo al usuario que ingresara una palabra con ayuda del gets posteriormente imprimí la palabra ingresada, con ayuda del while realice la siguiente condición si cadena en la posición cero era igual o diferente de 0, i se iva a incrementar, mando un mensaje diciendo lo que tenía la cadena al revés y con ayuda de un for pude voltear la cadena de manera inversa por ultimo mande a imprimir esta.



Se muestra la palabra introducida, se vuelve a imprimir la palabra introducida que es carmenperezsoto y después la cadena al revés que es otoszerepnemrac y efectivamente da lo que se esperaba.

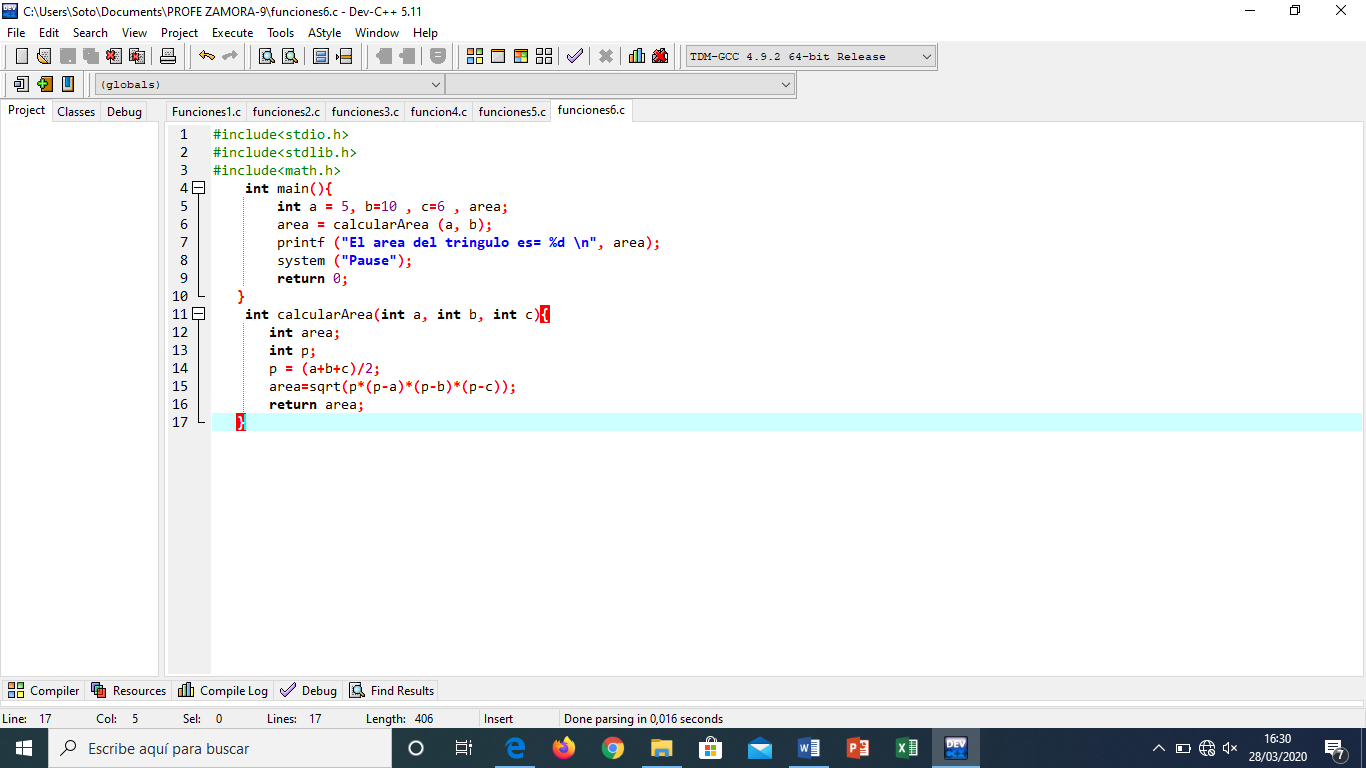


**Ejercicio 6**

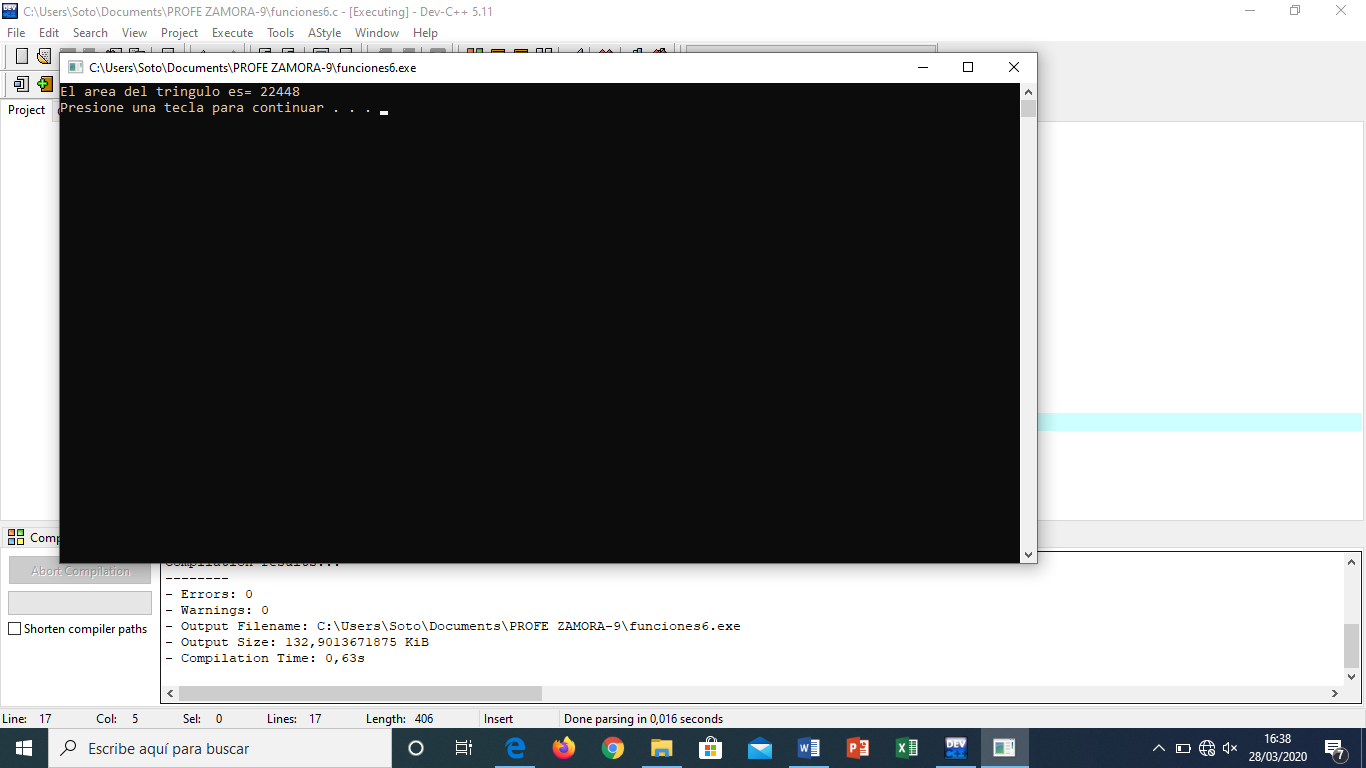


En este ejercicio lo que hice fue colocar le método principal dentro del coloqué 4 variables de tipo entero a=5, b=10 y c=6 que son los tres lados del triángulo y una variable llamada área que esta guardara el valor obtenido.

Después esta la variable área que esta igualada a lo que nos dará el método así que nos vamos a el para ver que nos está resolviendo, es un método de tipo entero llamado calcularArea con tres parámetros que son a, b y c con su tipo de dato. Dentro de su cuerpo estas dos variables de tipo entero una que es el araea y otra que va hacer p que nos ayudara a sumar los lados y dividirlos entre dos. Después esta la variable área y la fórmula para resolver y sacar el área, para sacar la raíz sube que usar la palabra sqrt y la librería #include<marth.h> que nos ayuda a sacar la raíz o funciones ya terminando se retorna lo que hay en la variable área y se imprime el método principal.



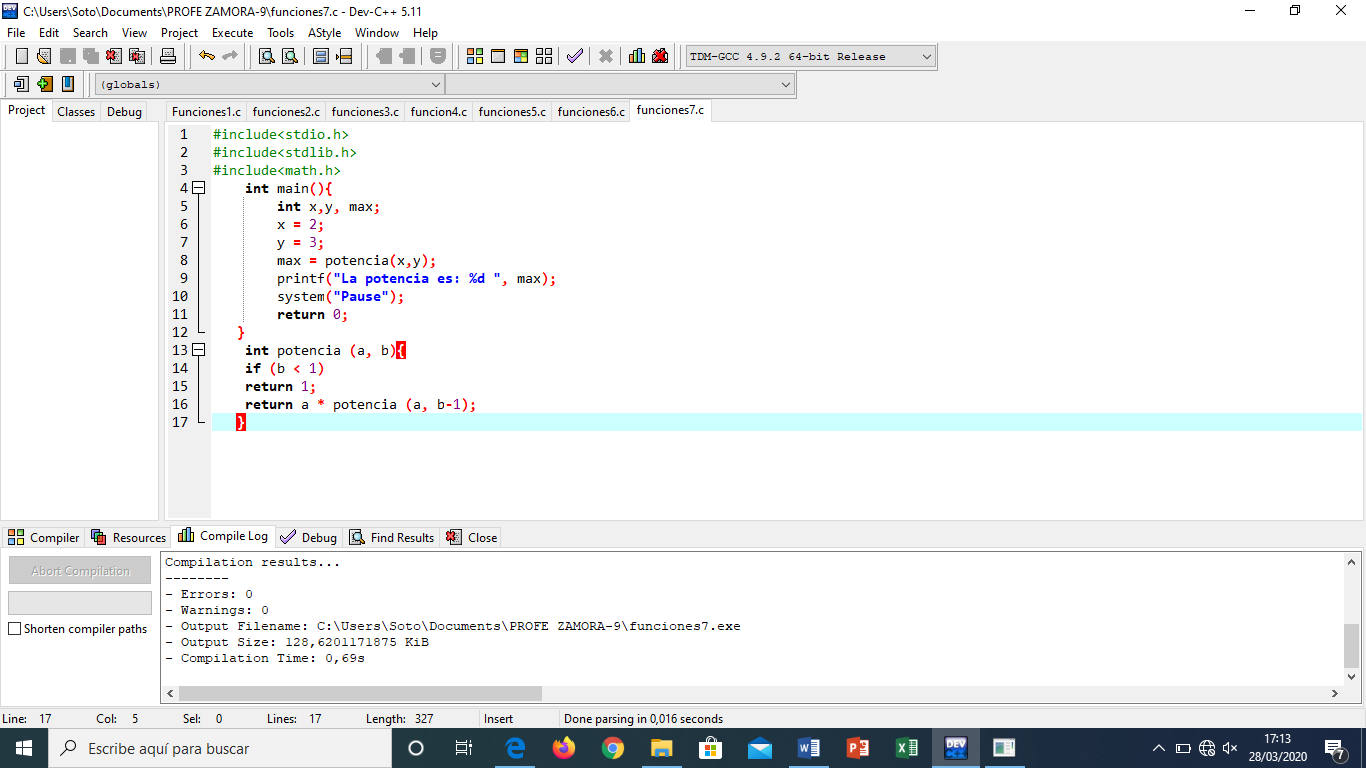
En la pantalla de abajo se muestra el resultado.

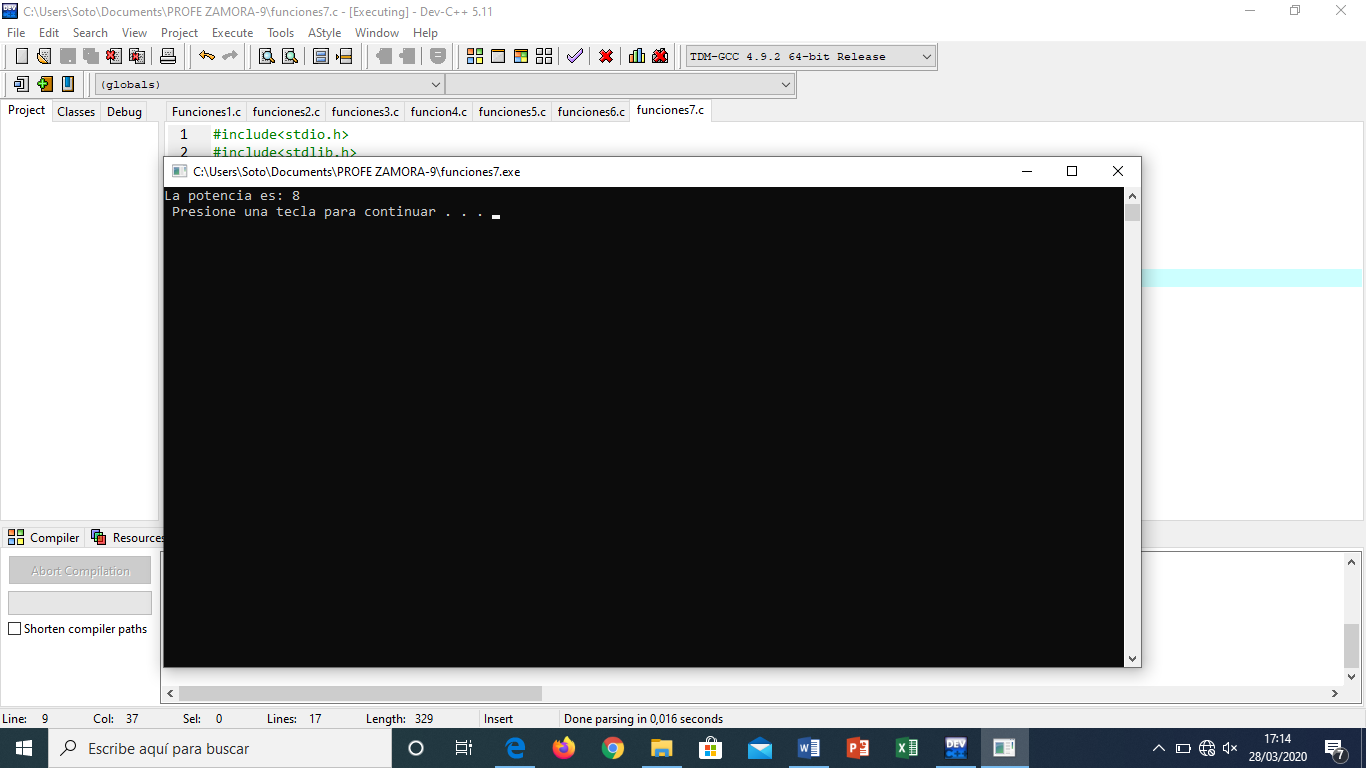


Se llaman funciones recursivas a aquellas que se llaman a su mismas de forma repetida hasta que se cumpla alguna condición.

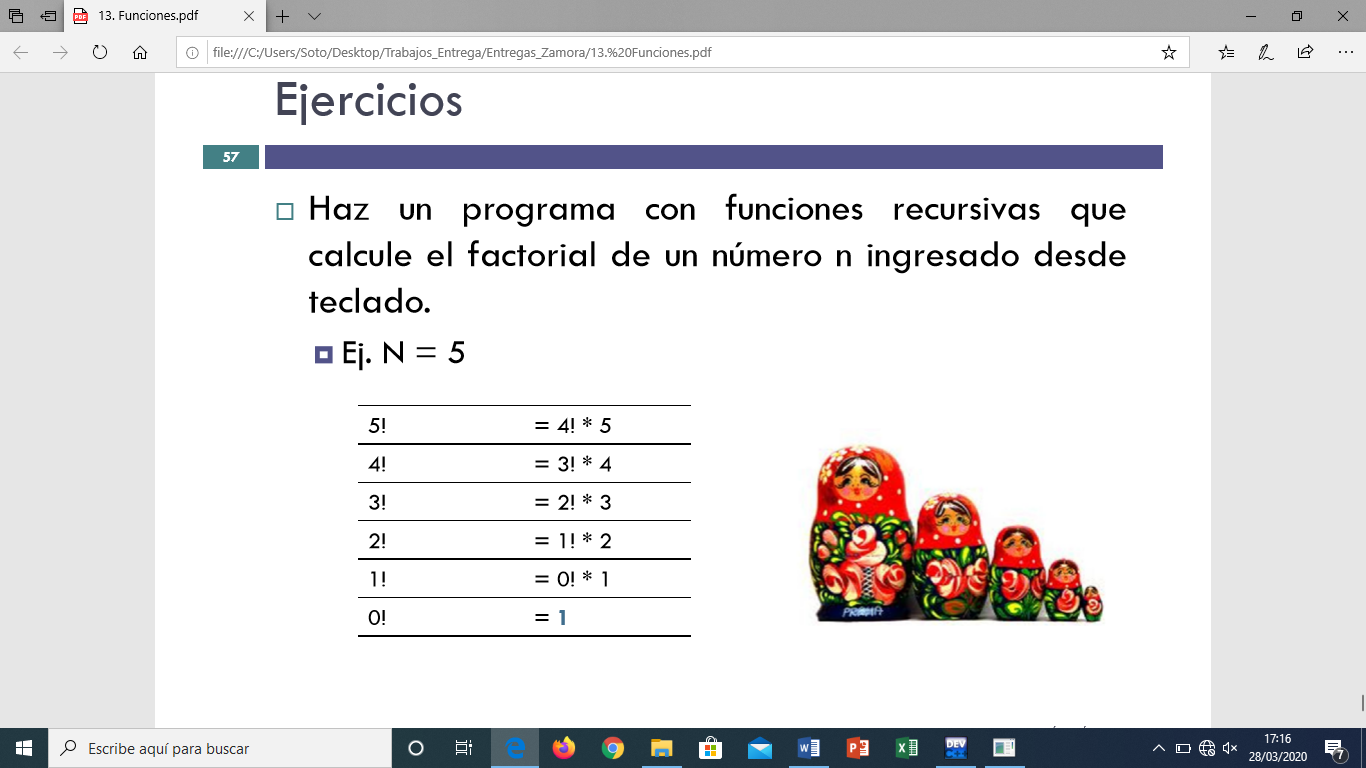
 Cada llamada implica el almacenamiento de variables de estado y otros parámetros

**Ejercicio 7**

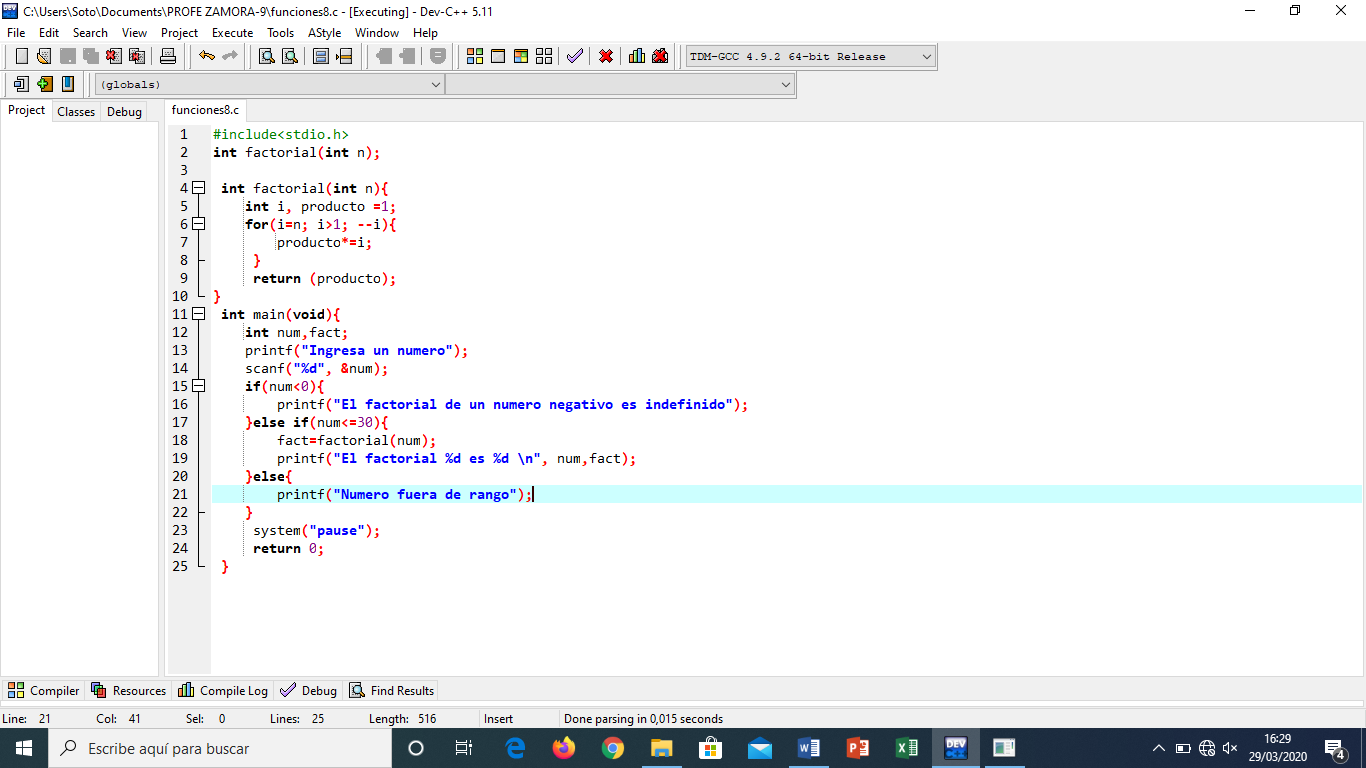




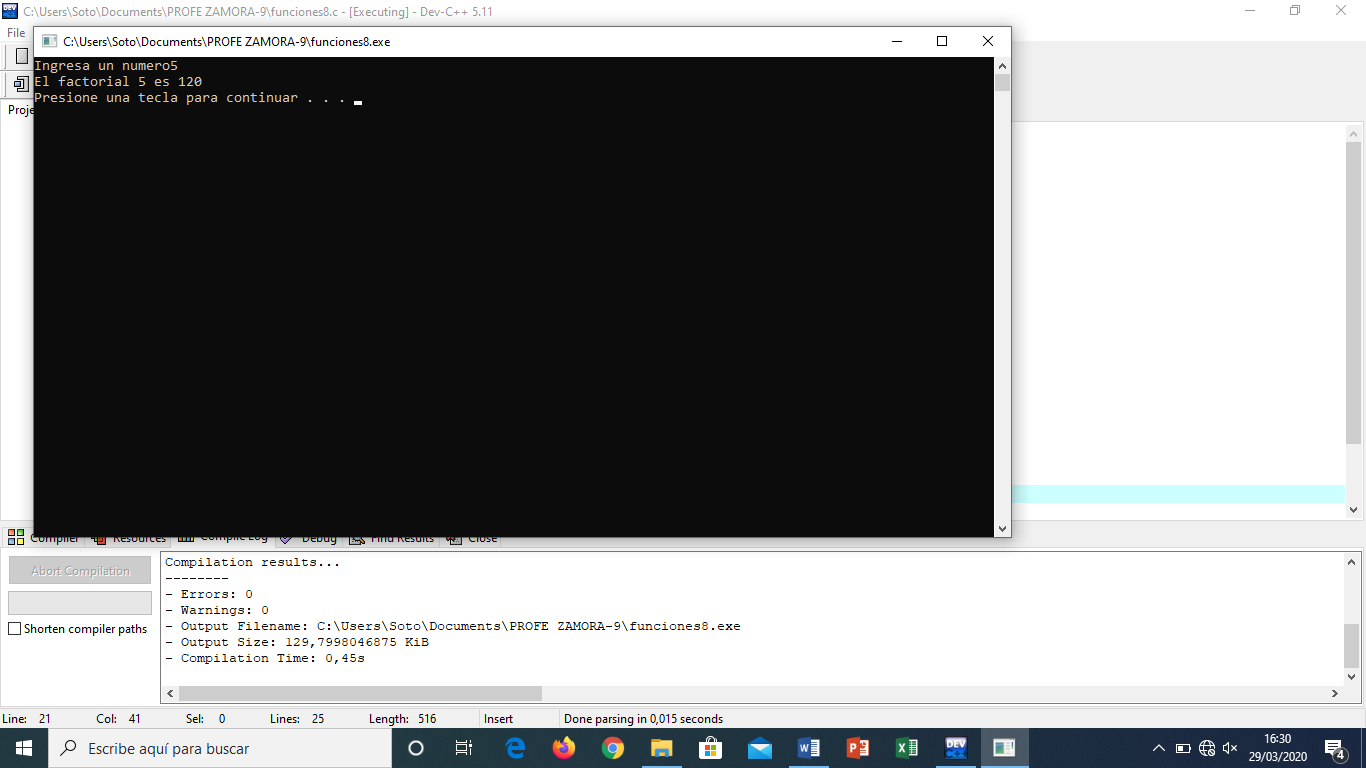
**Ejercicio 8**



En este ejercicio tuve que hacer dos métodos uno que sea la función y otra que es el método principal, el método factorial me ayudara a calcular el factorial del numero ingresado con ayuda de un for para poder calcularlo, el método principal me ayudara a pedir el numero el número y poder calcularlo.



En la pantalla de abajo se muestra el numero ingresado y su factorial.



**V. Conclusiones:** Estas laminas que realicé le entendí más a las cadenas con las funciones que tienen pues es algo más fácil, las funciones si se me complicaron porque hay que entender como es que funciona estas funciones nos ayudan a reducir código y poder usar una función para realizar un cálculo.