|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre de la práctica** | **CADENAS EN LENGUAJE C** | | | **No.** | **1** |
| **Asignatura:** | **MÉTODOS MUMÉRICOS** | **Carrera:** | **ISIC** | **Duración de la práctica (Hrs)** | **2** |

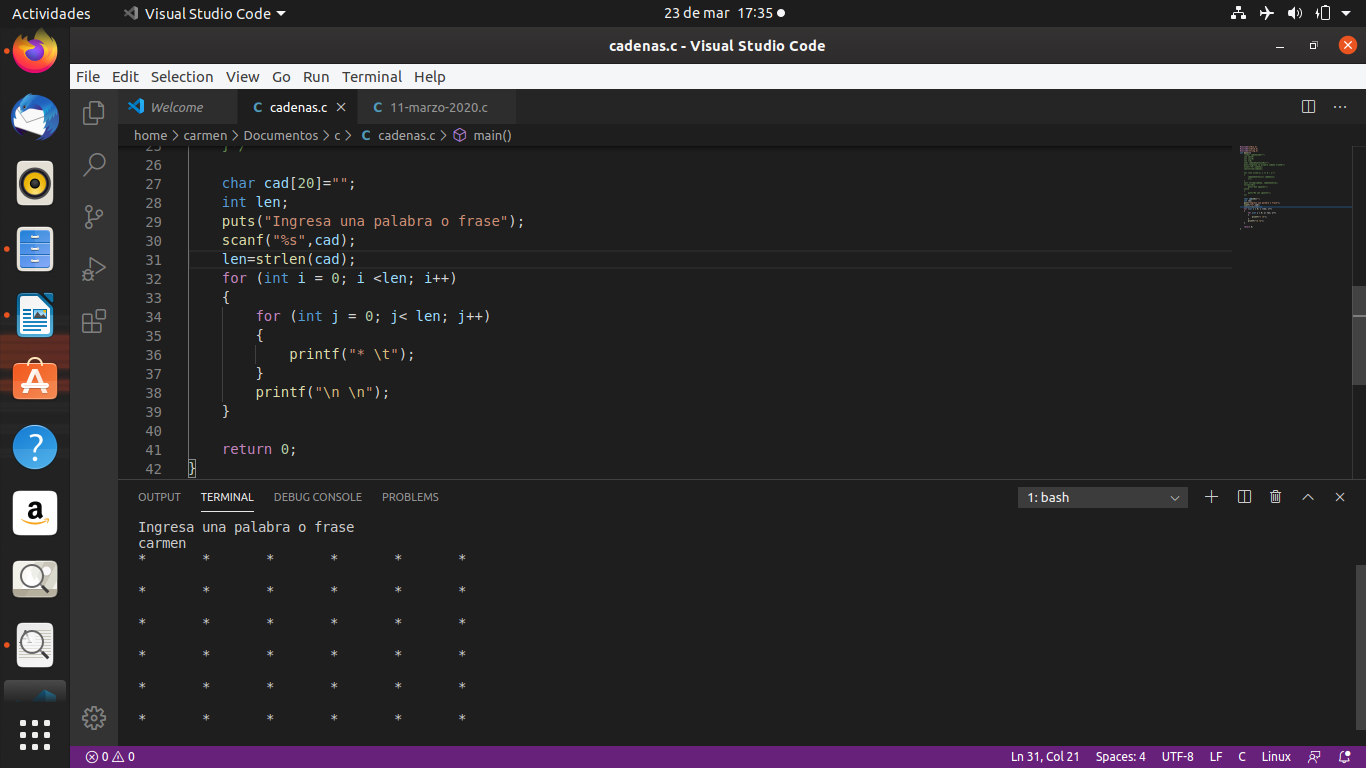
**I. Competencia(s) específica(s):\*\*\*\*\*\*\***

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

**II. Desarrollo de la práctica:**

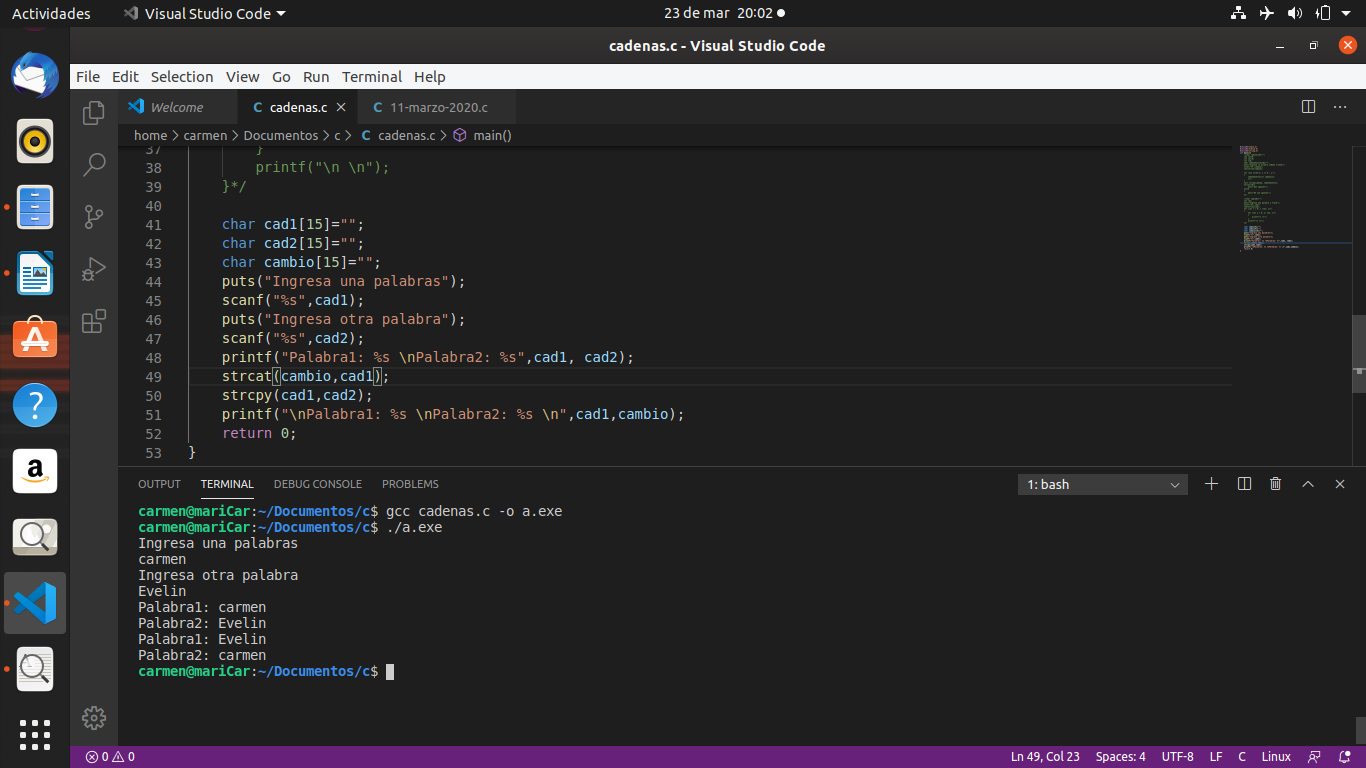
**CADENAS**

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



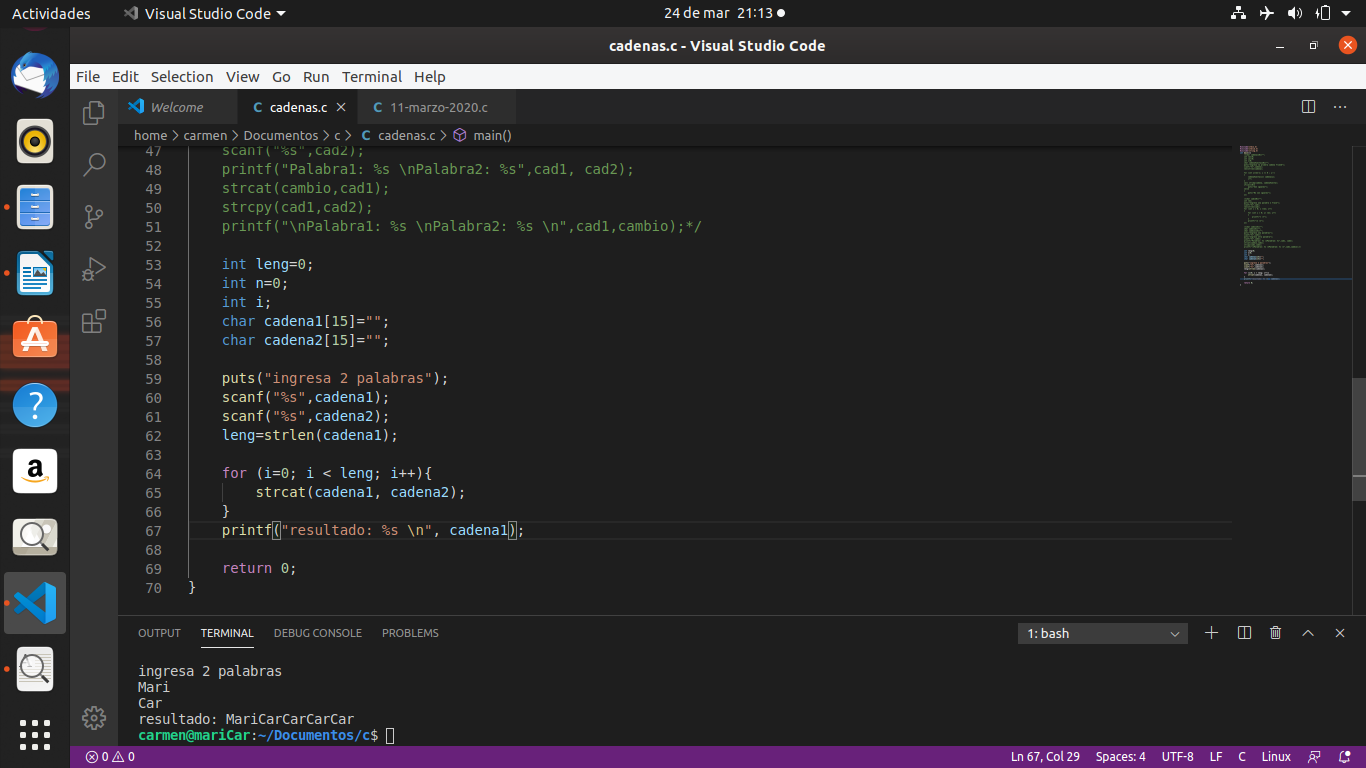
* Para su solución utilizamos arreglos de tipo Char para almacenar la palabra que el usuario ingresara
* **puts y scanf:**  estos los utilizamos para imprimir el mesaje y almacenar la palabra que ingresara el usuario en la variable creada.
* **Strlen:**  Nos ayuda a obtener el valor numérico de caracteres que tiene la cadena.}
* el valor numérico que obtendrá nos ayudara para poder hacer las sentencias de repetición.
* El primer ciclo for nos ayuda a ser las iteraciones para que pueda imprimir el número de columnas y el segundo for para imprimir el numero de asteriscos en cada fila.

1. 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



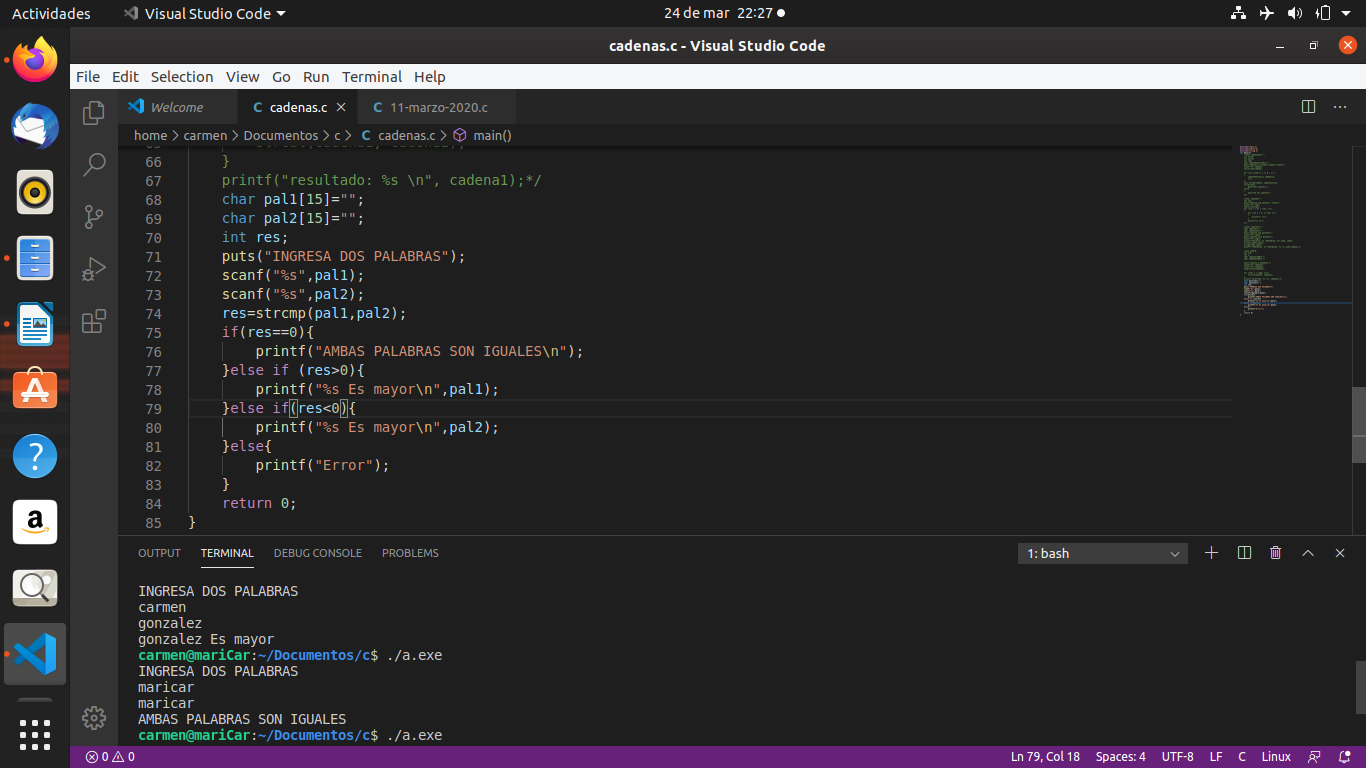
* creamos 3 arreglos, 2 de ellos para cada una de las palabras que ingresará el usuario
* **puts**: para imprimir un mensaje en consola
* **scanf:** para pedir datos y almacenarlos en una variable, en este caso en un arreglo
* la herramienta **strcpy**  nos ayudará para compiar el contenido de la cadena origen a la cadena destino.
* Y la herramienta **strcat:**  para concatenar la **cad1** a la cadena **cambio** en el código
* finalmente imprimiremos los resultados

1. 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.



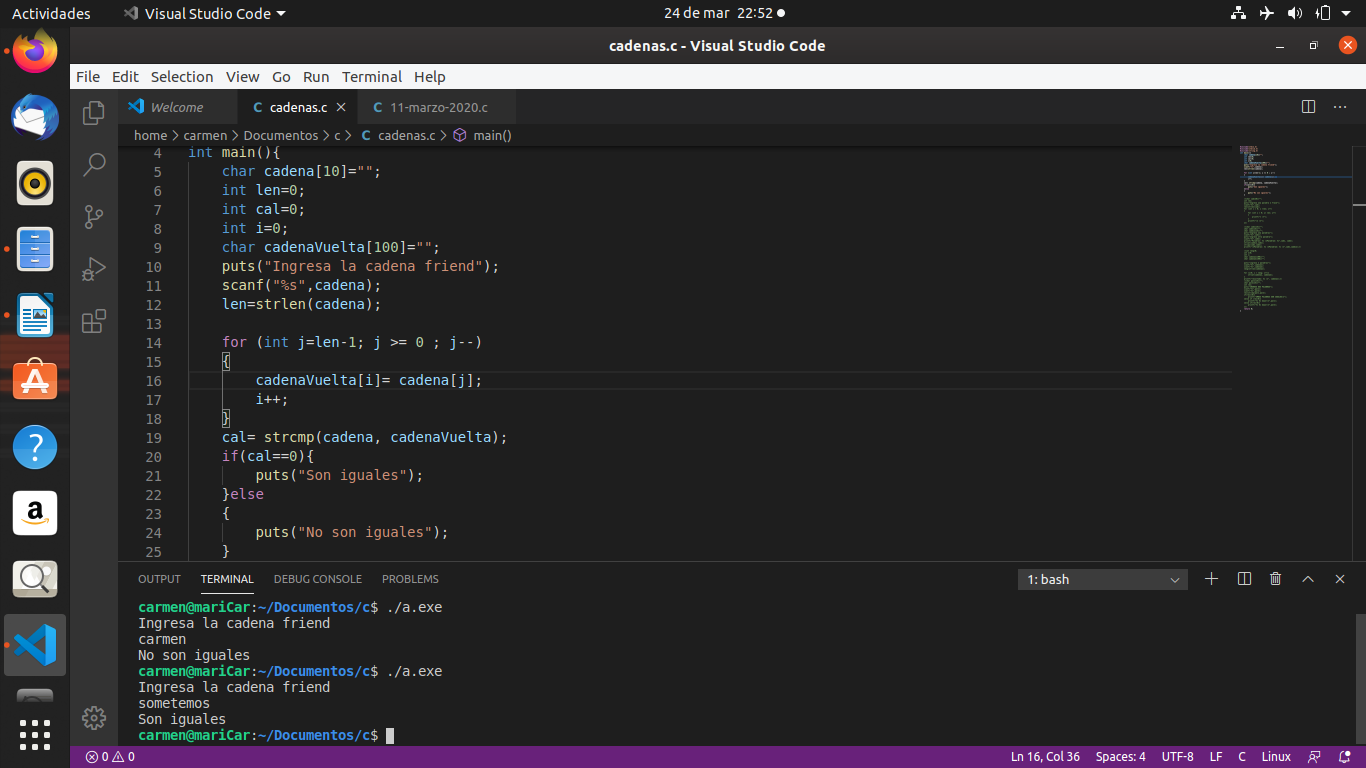
* Declaramos 2 arreglos para almacenar las palabras que el usuario ingresara, 2 variables mas de tipo entero la de **leng** para obtener el numero de caracteres que tiene la primera palabra que el usuario ingresara, esto lo obtendremos con ayuda de la herramienta **strlen.**
* **Scanf:** para almacenar las palabras que ingresara el usuario
* El ciclo **for**  nos ayudará para acumular el numero de repeticiones de la segunda palabra
* Finalmente imprimiremos la **cadena1** en donde se acumularon las palabras

1. Escribe un programa que reciba por teclado dos palabras y te indique cual de ellas es mayor y cual es la menor. En caso de ser iguales, que imprima la leyenda 'ambas palabras son iguales'.



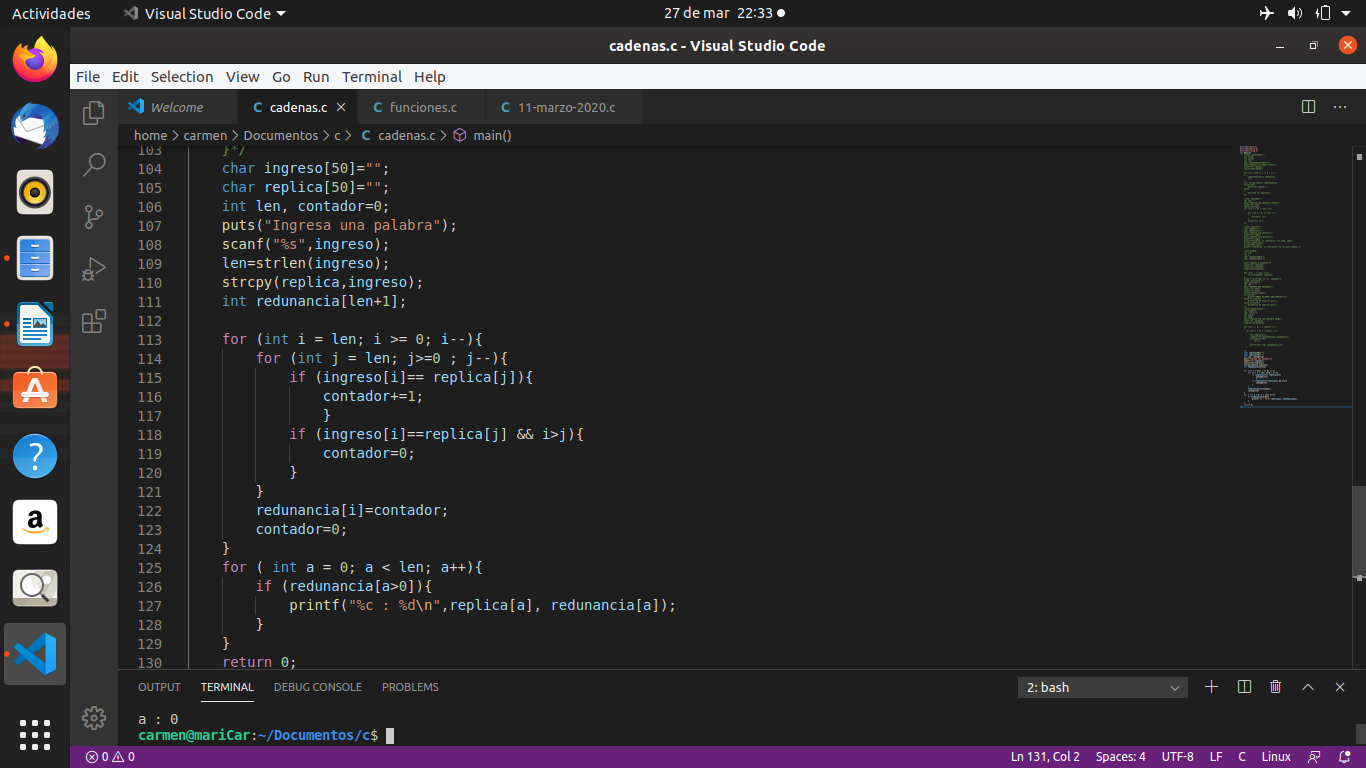
* **puts:** Para enviar un mensaje.
* **scanf:**  utilizamos para pedir datos por consola y almacenarlos en una variable, en este caso en un arreglo ya que fueron palabras las que ingresara el usuario.
* **strcmp:** Lo utilizamos para obtener un valor numérico que nos indicara la comparación de las palabras; **0 si las palabras son iguales, un numero negativo si la cadena1 es menor que la cadena2.**
* **if, else if:**  Lo utilizamos para enviar los mensajes de acuerdo al resultado que obtuviera el **strcmp.**

1. Crea un programa que detecte una palabra palíndroma.



* Declaramos 2 arreglos una para almacenar la palabra que el usuario ingresara y la otra para la misma palabra pero volteada, esto para su comparación.
* 3 variables de tipo entero, la primera sera para obtener la cantidad de caracteres que tiene la palabra ingresada por el usuario y la ultima para almacenar el resultado que nos enviara el **strcmp.**
* **puts:** para imprimir un mensaje
* **scanf:**  para pedir la palabra al usuarioy almacenarla en el arreglo de tipo char que hemos creado.
* **strlen:** Para obtener el numero de caracteres que tiene la palabra y almacenarla en una variable.
* **For:** En este ciclo lo que hicimos es almacenar la cadena ingresada por el usuario Alrevez en el arreglo **cadenaVuelta**, para posteriormente comparar estas 2 cadenas (**strcmp**) y obtener un valor que nos resolverá si la palabra es igual enviara un 0 de lo contrario no es igual. Esto se decidirá con un sentencia condicional **if.**

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



* Declaramos 2 cadenas
* en la tercer linea declaramos las variables que vamos a ocupar, **len:(strlen)** para obtener el numero de caracteres de la cadena y un **contador**
* **scanf:** Para pedir datos de teclado
* **strcpy:** Copea el contenido de una cadena a otra
* declaramos una variable que almacenara las ocurrencias de las letras en la palabra
* Los ciclos anidados funcionan de la siguiente manera:

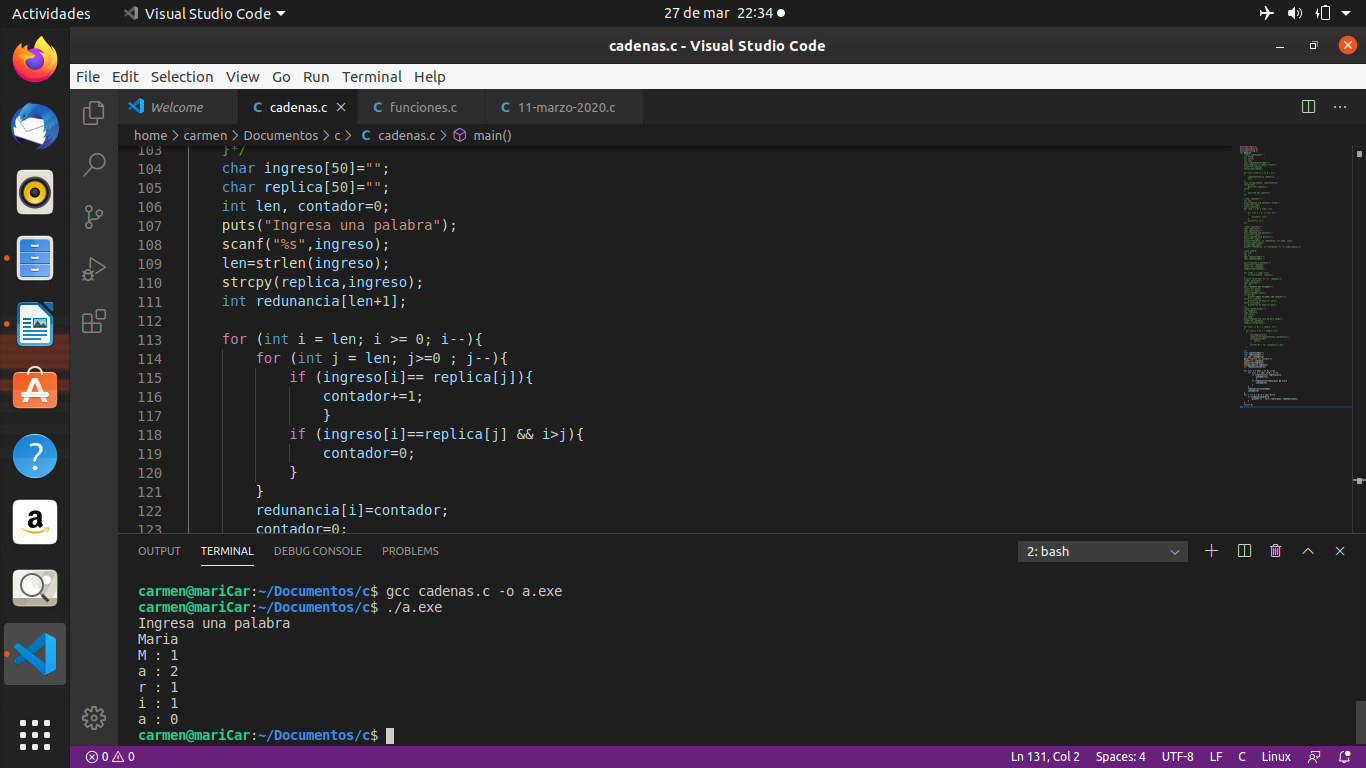
**1erfor:**  la variable **i** tendrá como valor el numero de caracteres de la palabra que hemos obtenido con la herramienta **strlen.** Este ciclo nos ayuda para estabilizar una posicionen la palabra

**El segundo for:**  La variable **j**  tendrá el mismo valor que **i.** Este ciclo recorrerá con la posicion de i en la palabra cada una de las letras, comparándolas y en caso de que la letra se repita aumentara una unidad al contador que sera el numero de redundancias de la letra en la palabra.

Cuando terminen las iteraciones del segundo ciclo, se colocara el numero de redundancias en el arreglo  **redundancia**  en la posicion **i**  que fue la posicion de la letra que fue comparada. Así sucesivamente hasta que la variable **i**  sea menor a  **0**  o igual a **-1 ,** por que en ese momento ya no existe tal posición el en arreglo.

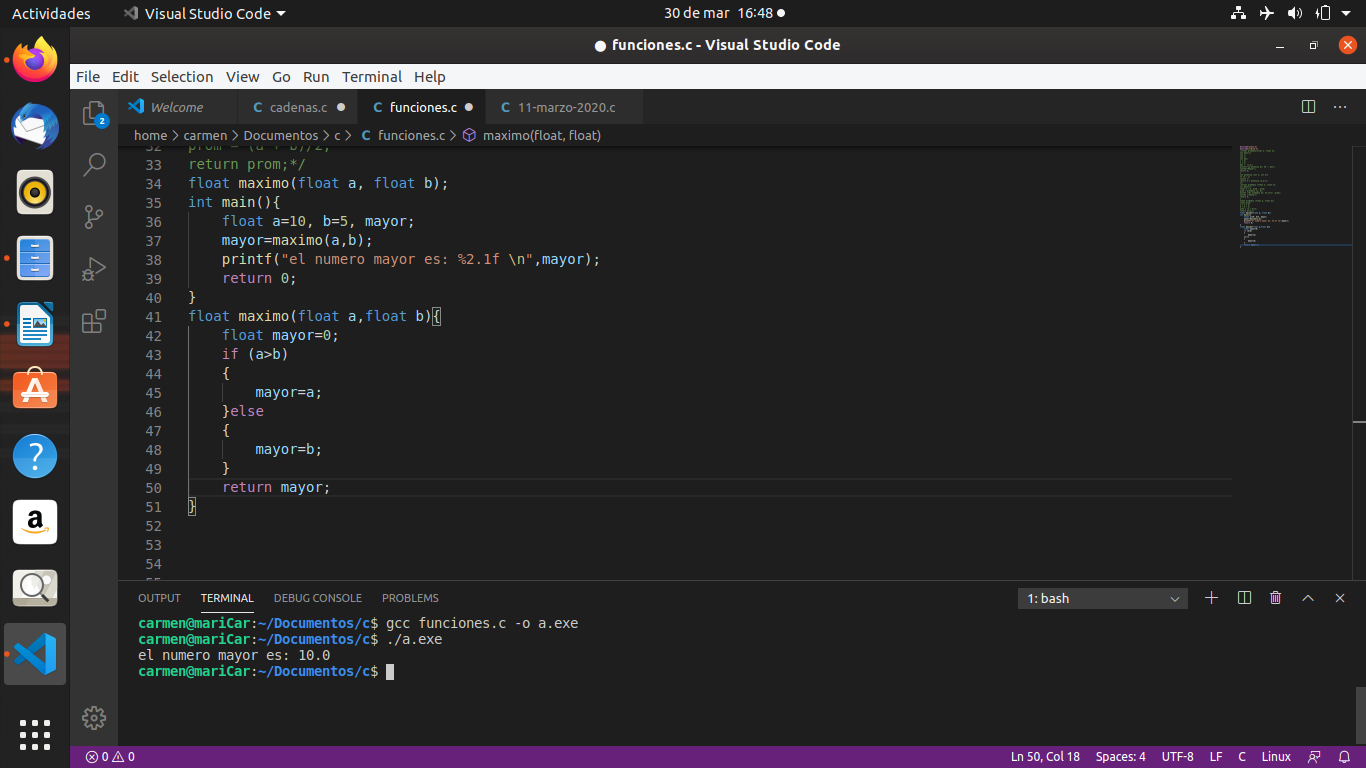
Por ultimo con ayuda de otro arreglo imprimiremos los resultados.

* **Resultado:**

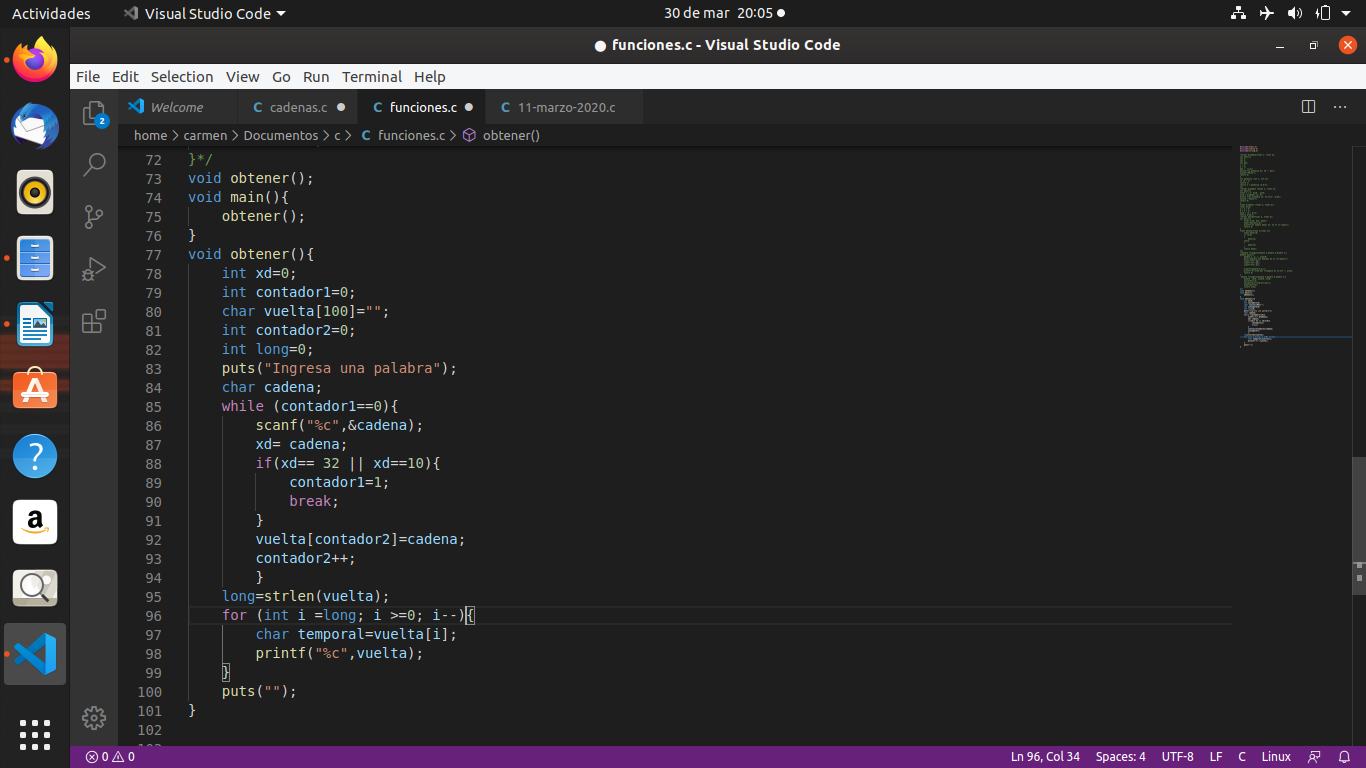


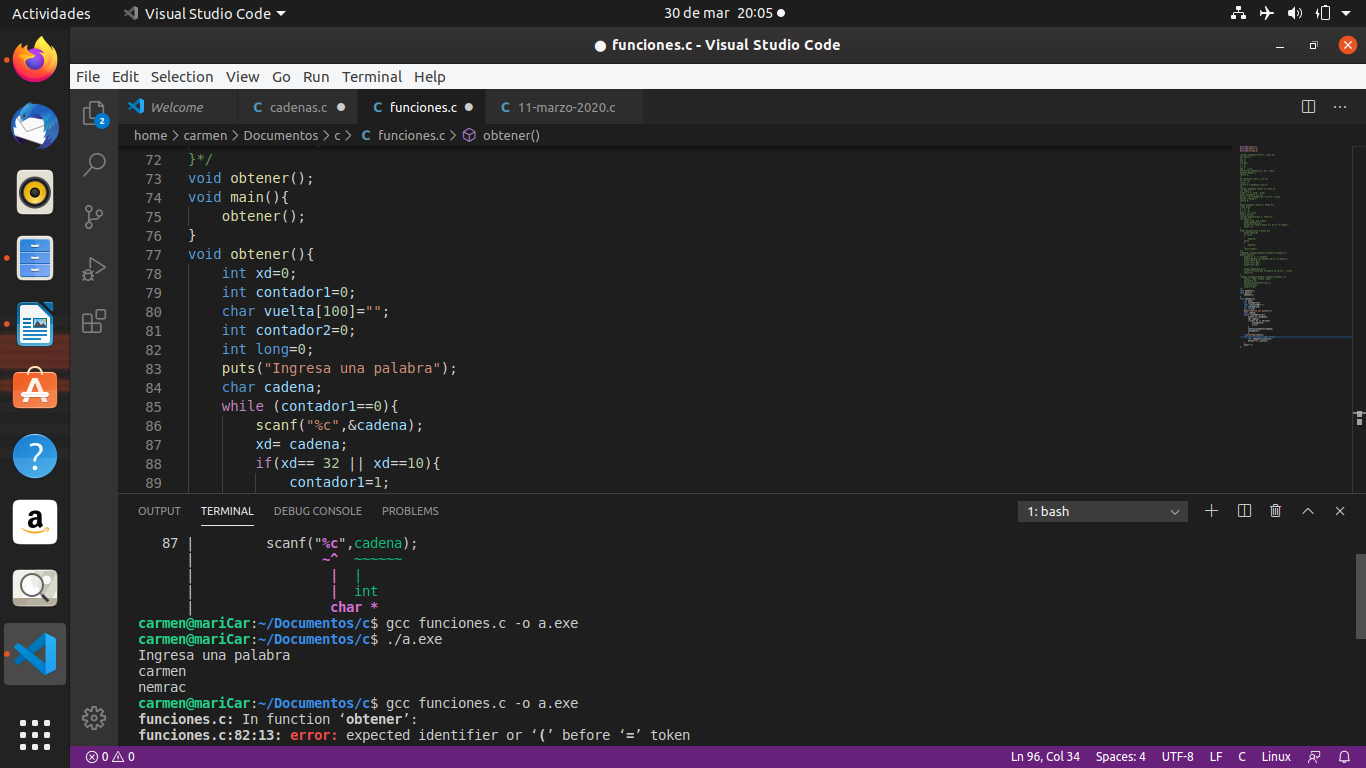
**FUNCIONES**

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

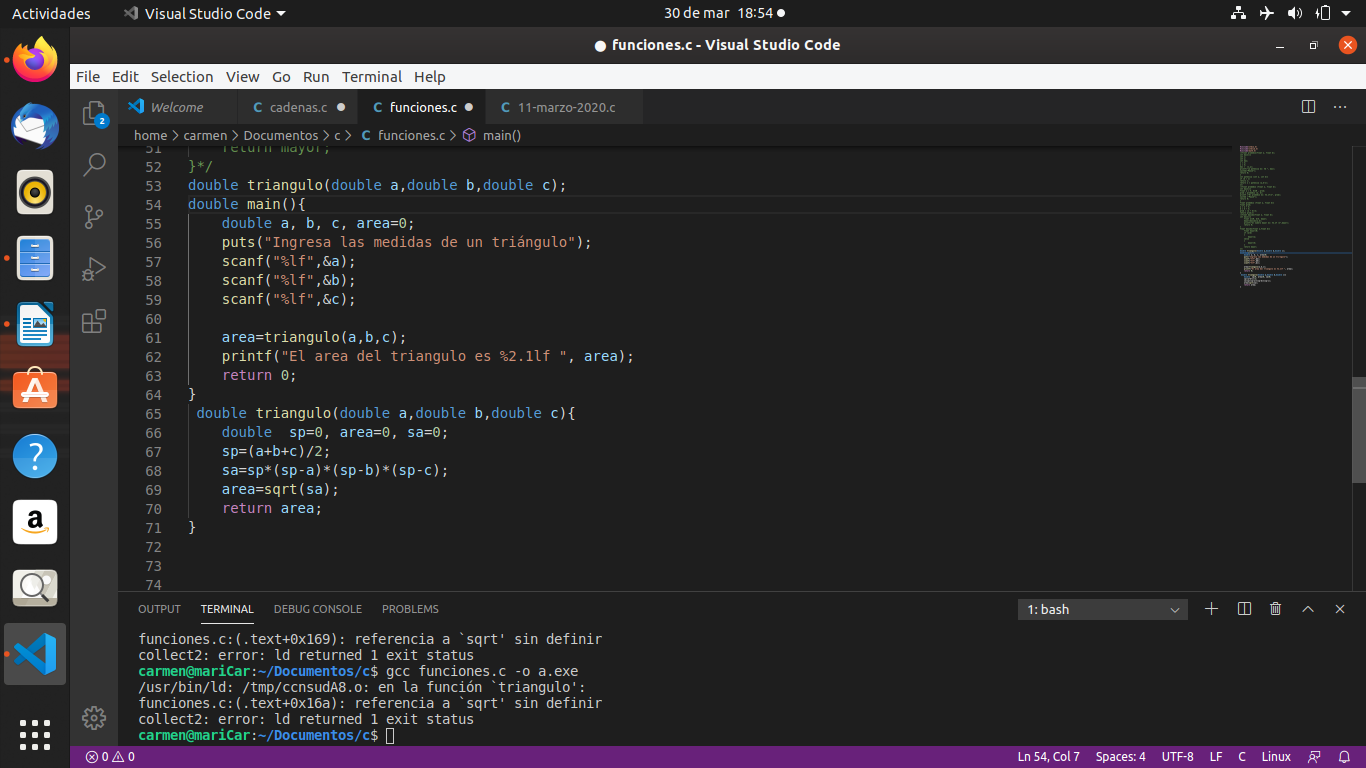


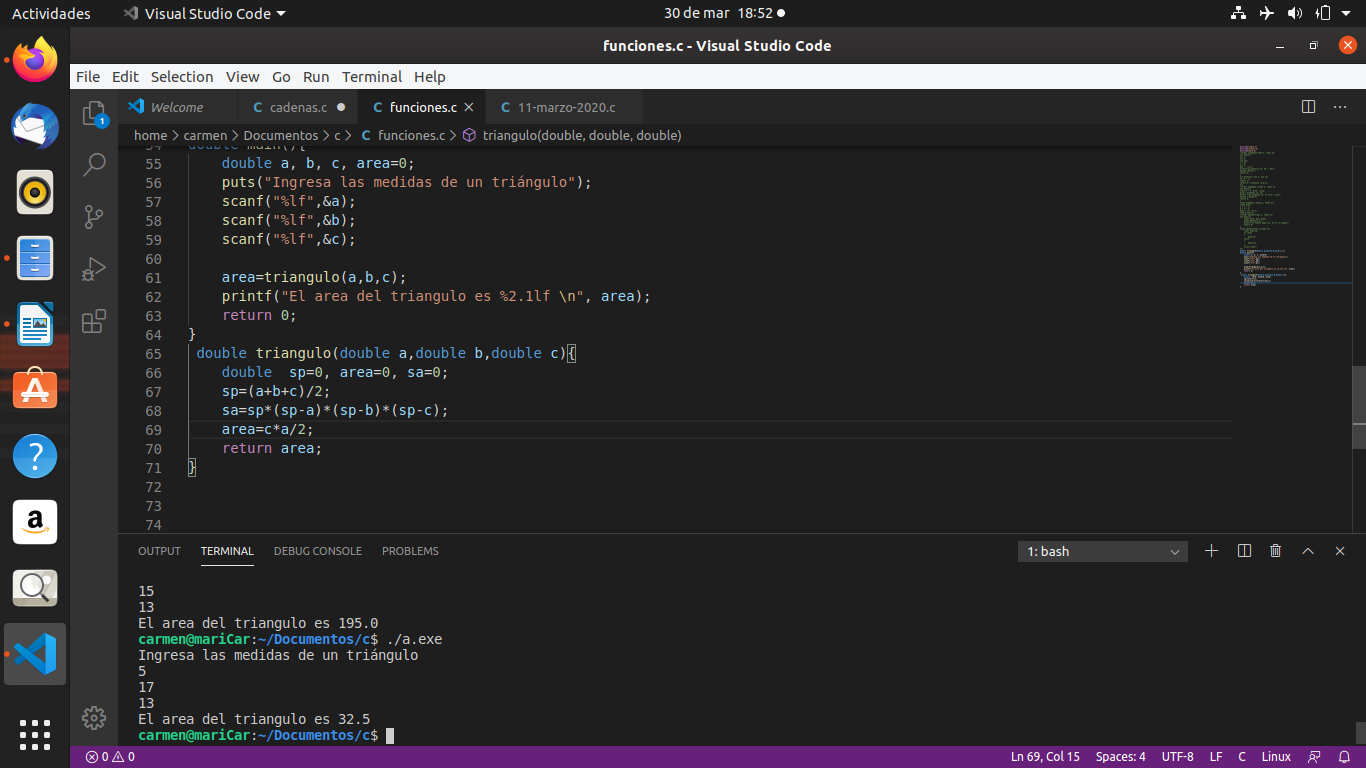
Escribir una función que reciba caracteres del teclado hasta recibir un espacio o un salto de línea (enter) y a continuación mostrar todos los caracteres en orden inverso.





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





**IV. Conclusiones:**