|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre de la práctica** | **Cadenas y Funciones en C** | | | **No.** | **3** |
| **Asignatura:** | **Métodos Numéricos** | **Carrera:** | **ISIC** | **Duración de la práctica (Hrs)** | **6 Hrs** |

**Alumno:**

* **Maria del Rosario Valentín Montiel**
* **3042**

**I. Competencia(s) específica(s):**

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

* hogar

**III. Material empleado:**

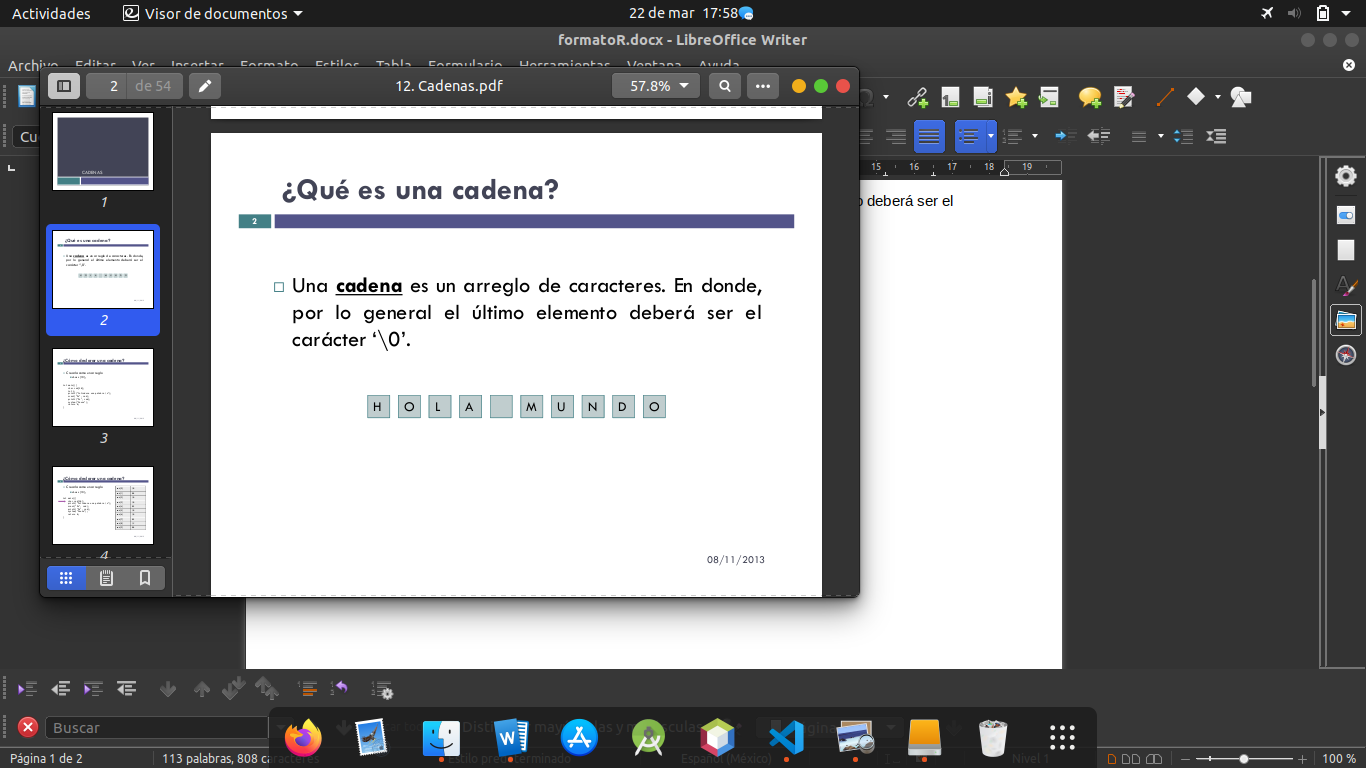
* lenguaje C
* Visual Studio Code

**IV. Desarrollo de la práctica:**

**CADENAS**

Una cadena es un arreglo de caracteres. En donde, por lo general el último elemento deberá ser el

carácter ‘\0’.

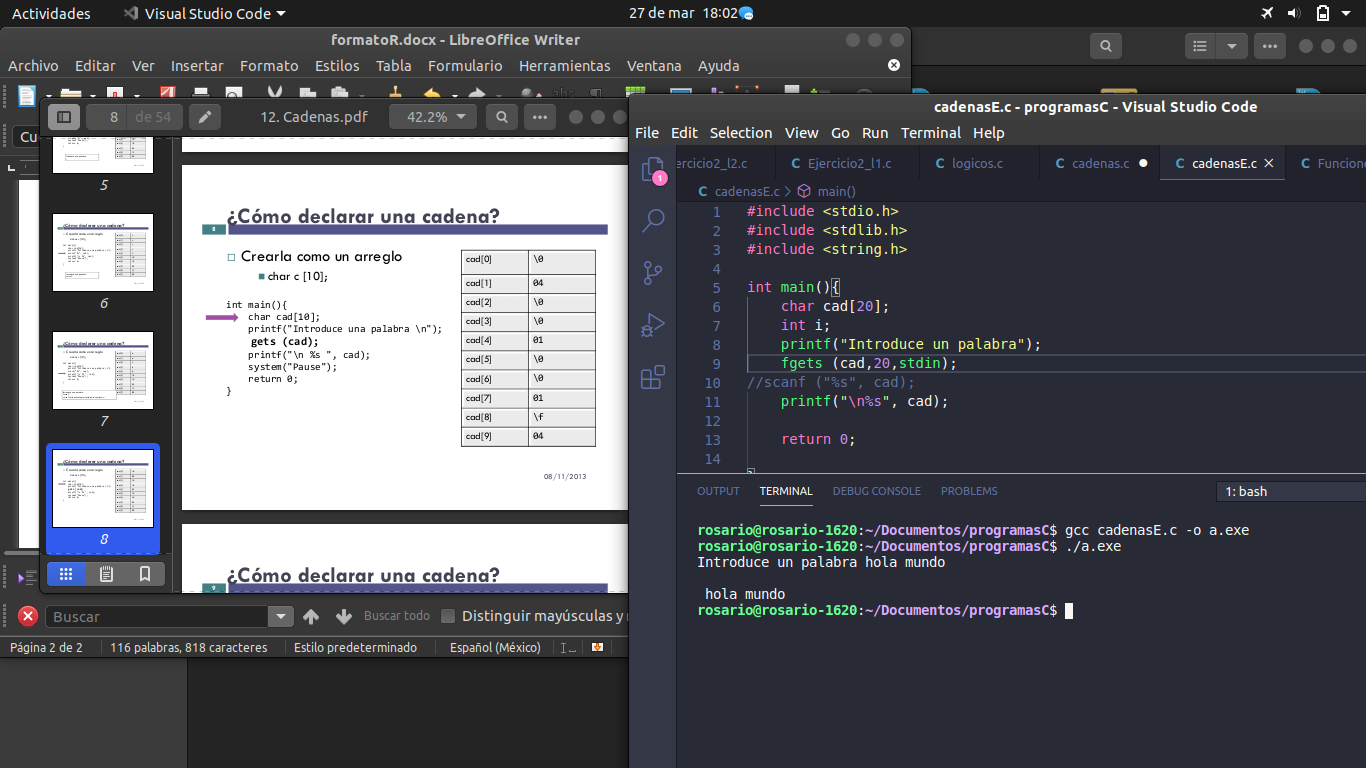


* ***EJEMPLO 1***



* **EJEMPLO 2**

concatena la cadena

****

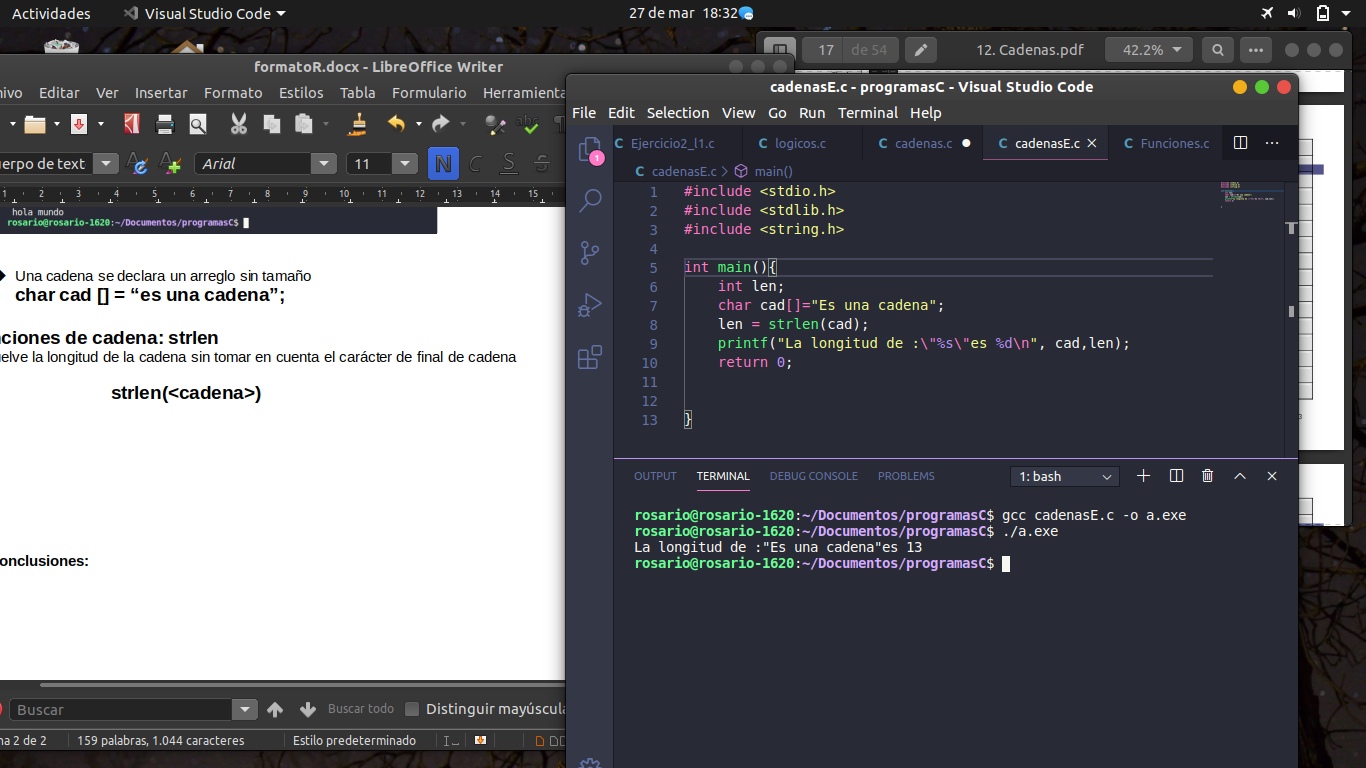
* Una cadena se declara un arreglo sin tamaño

**char cad [] = “es una cadena”;**

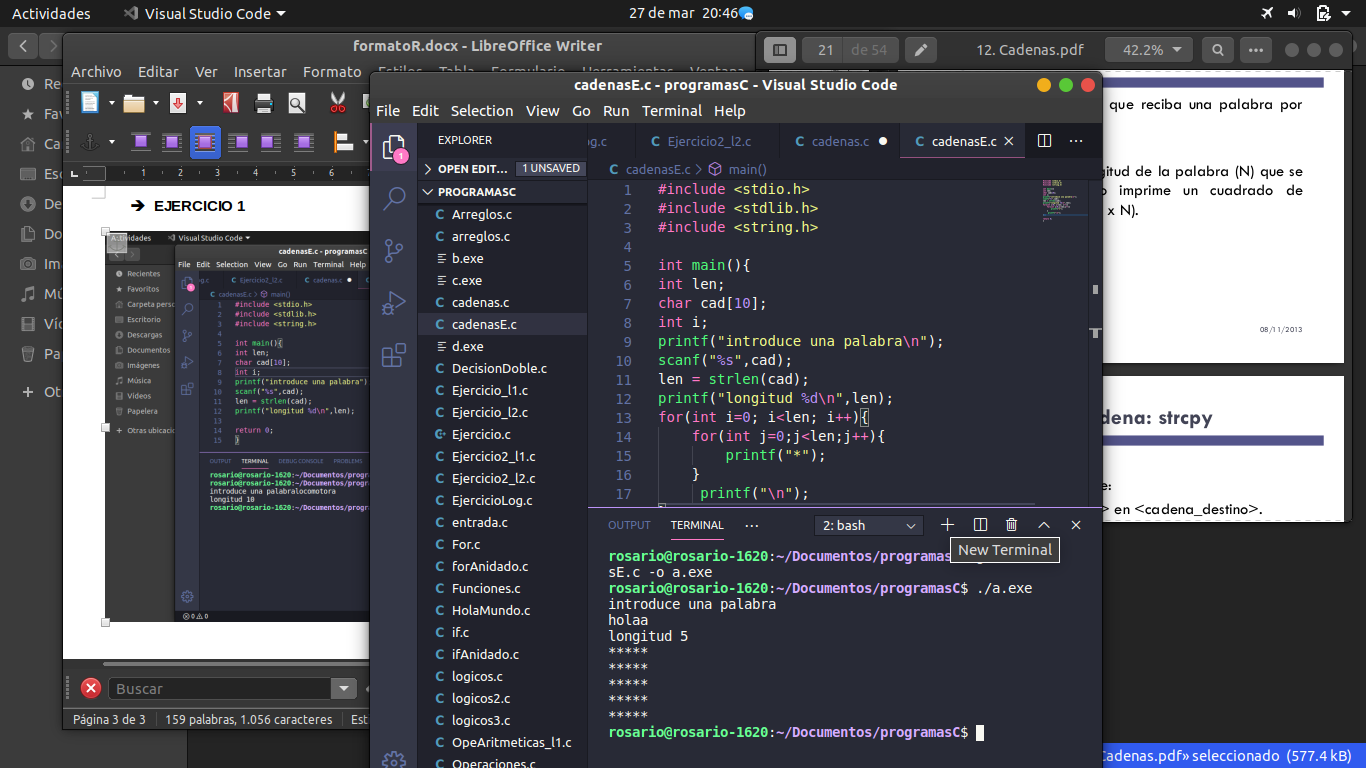
**Funciones de cadena: strlen**

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

**strlen(<cadena>)**



* **EJERCICIO 1**
* 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).

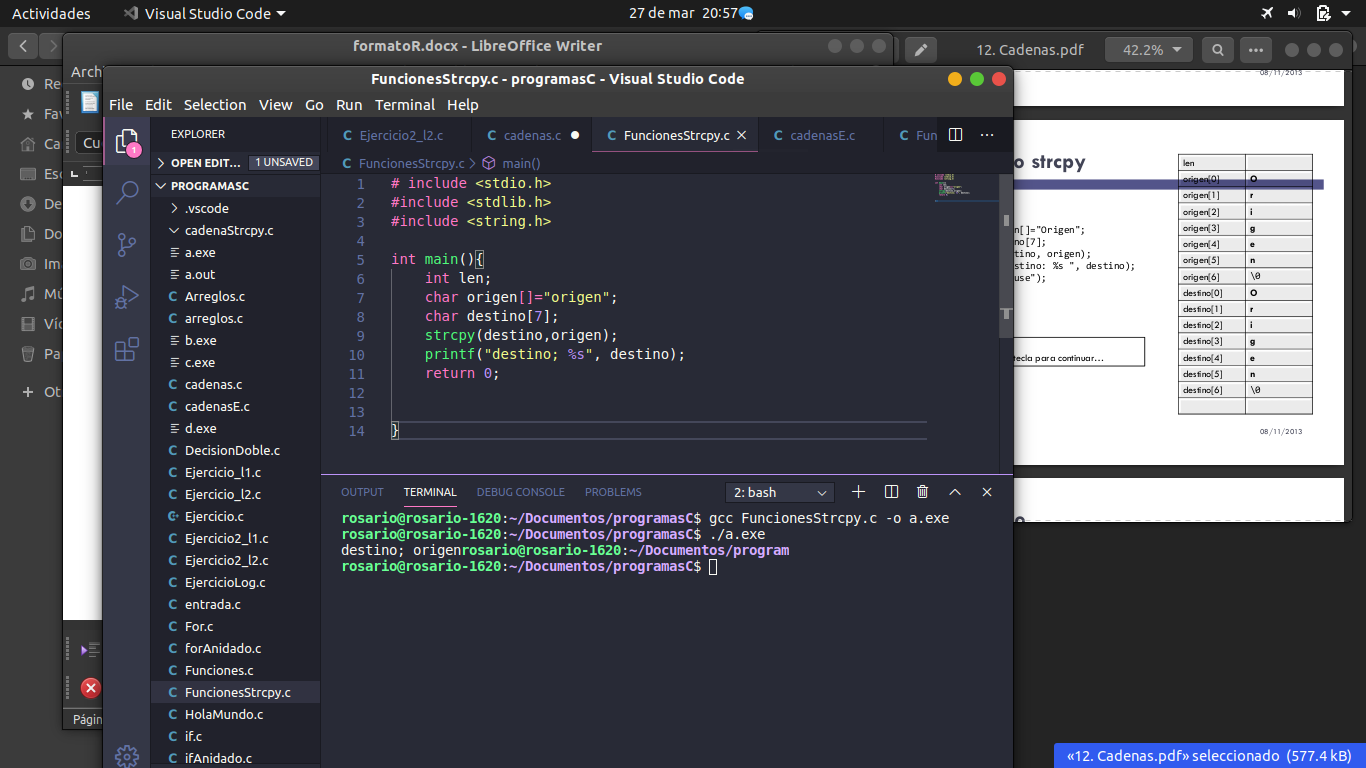
****

**Funciones de cadena: strcpy**

copia el contenido de :

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

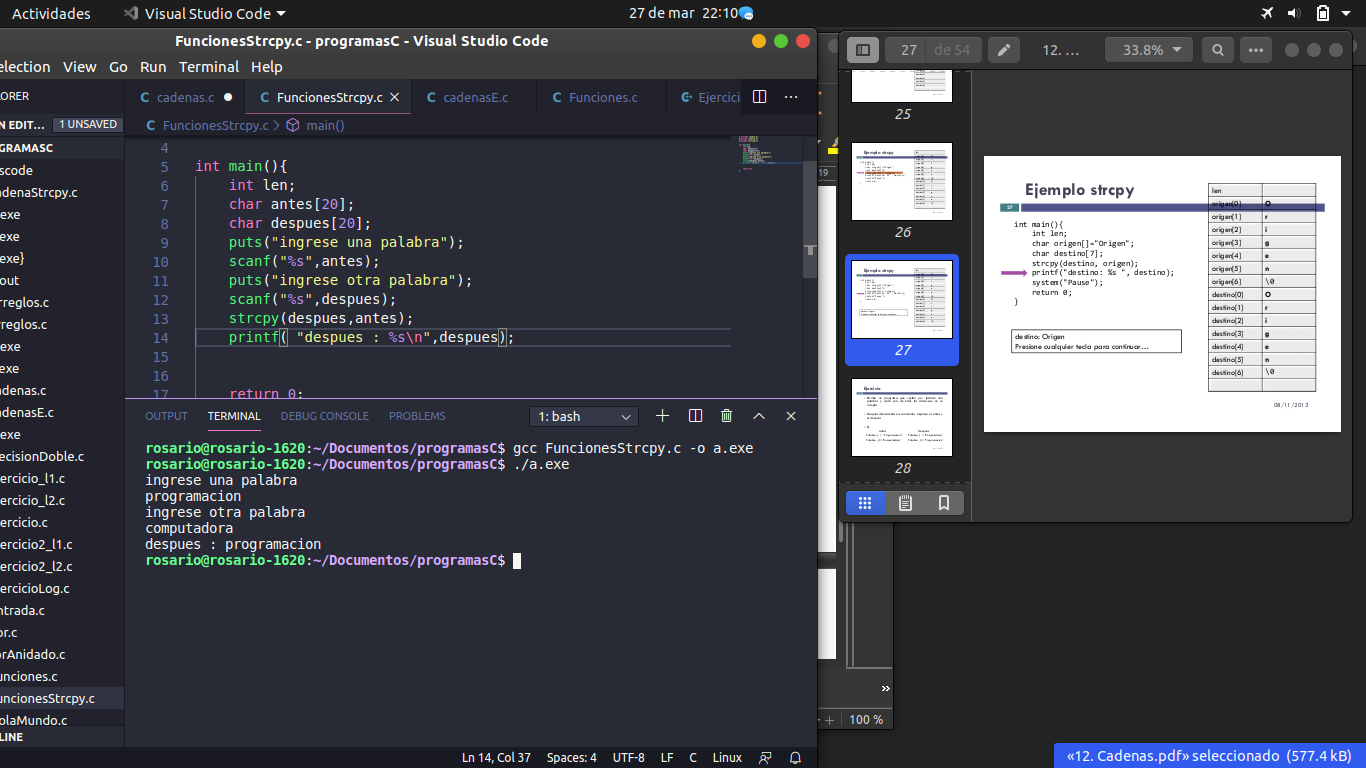
* **EJEMPLO 1**

****

* 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

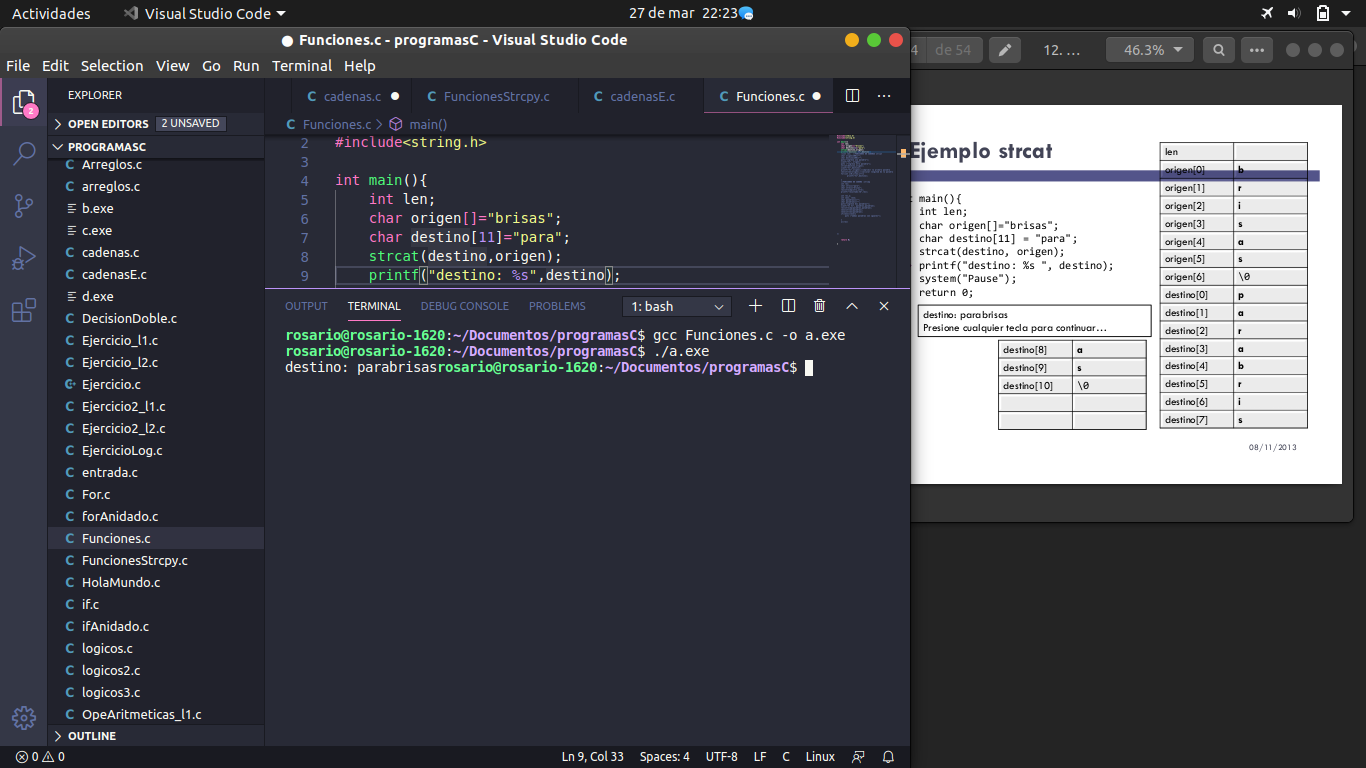


**Funciones de cadena : strcat**

Concatena el contenido de **<cadena\_origen> alfinal de <cadena\_destino>**

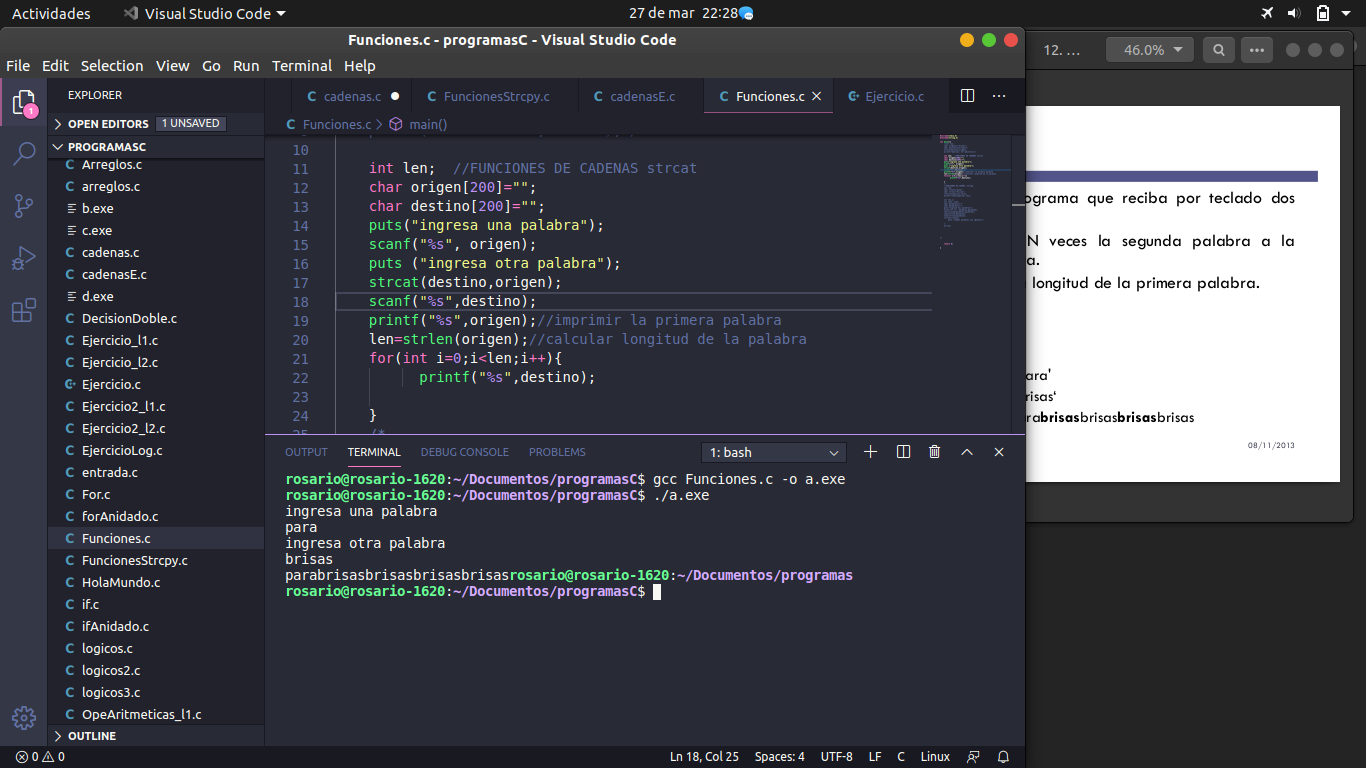
**strcat(<cadena\_destino>, <cadena\_origen>)**

* **Ejemplo 1**

****

**EJERCICIO 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.

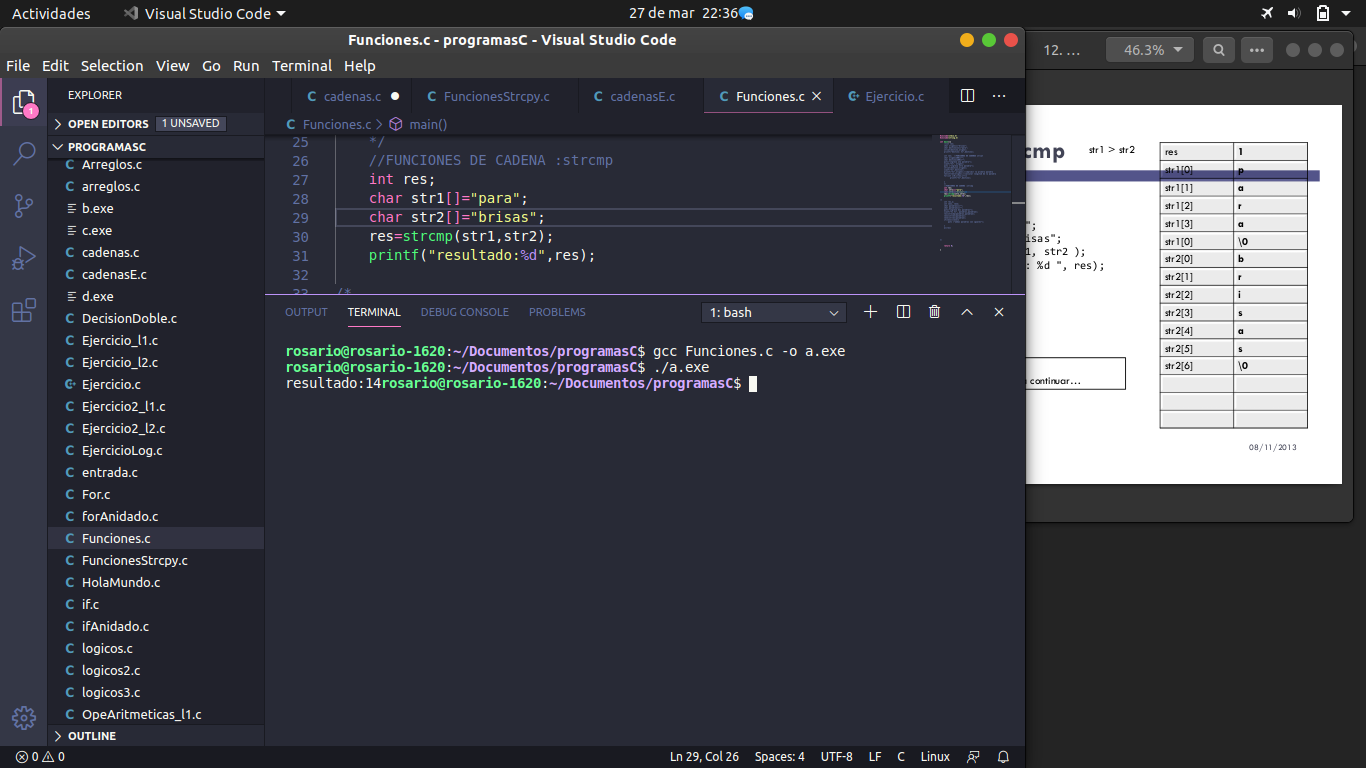
****

**Funciones de cadena: strcmp**

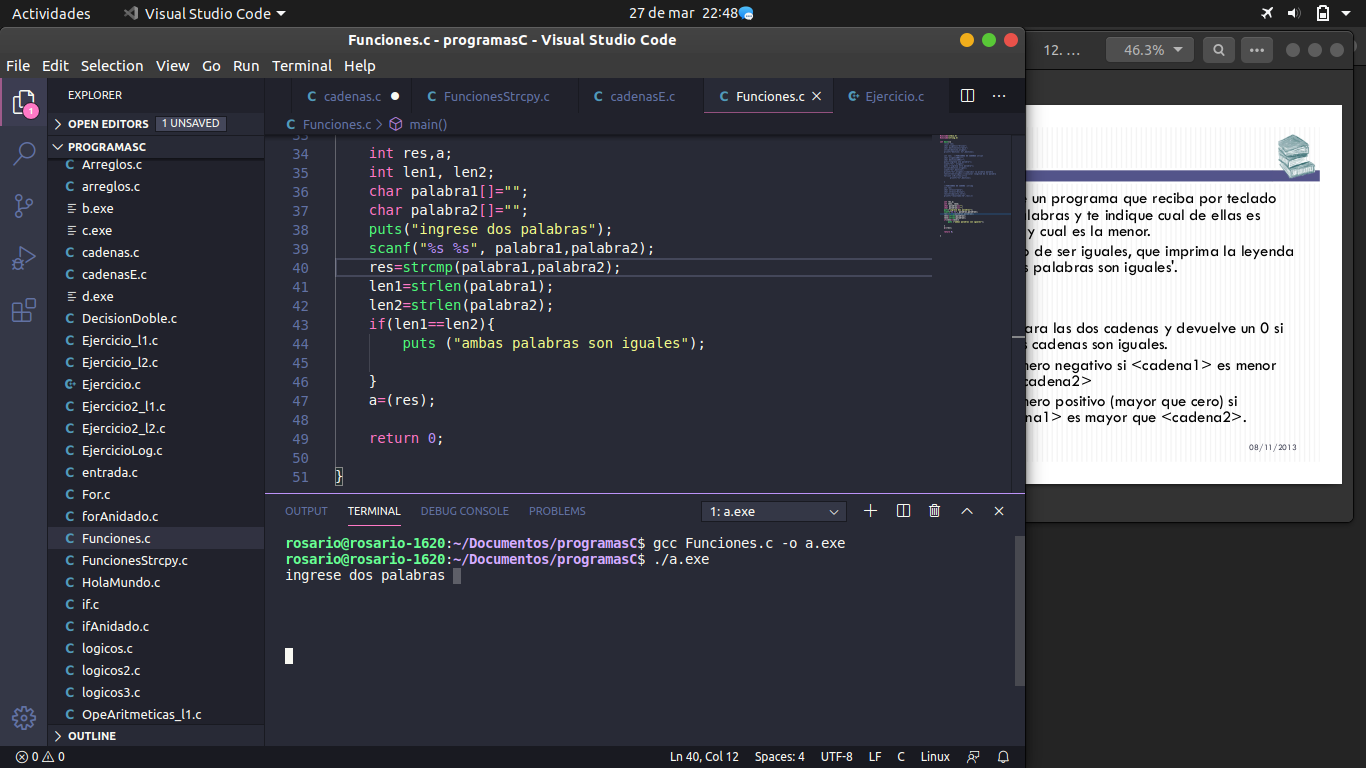
* Compara las dos cadenas y devuelve un 0 si las dos cadenas son iguales.
* Un número negativo si <cadena1> es menor que <cadena2>
* Un número positivo (mayor que cero) si <cadena1> es mayor que <cadena2>.

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

**EJEMPLO 1**

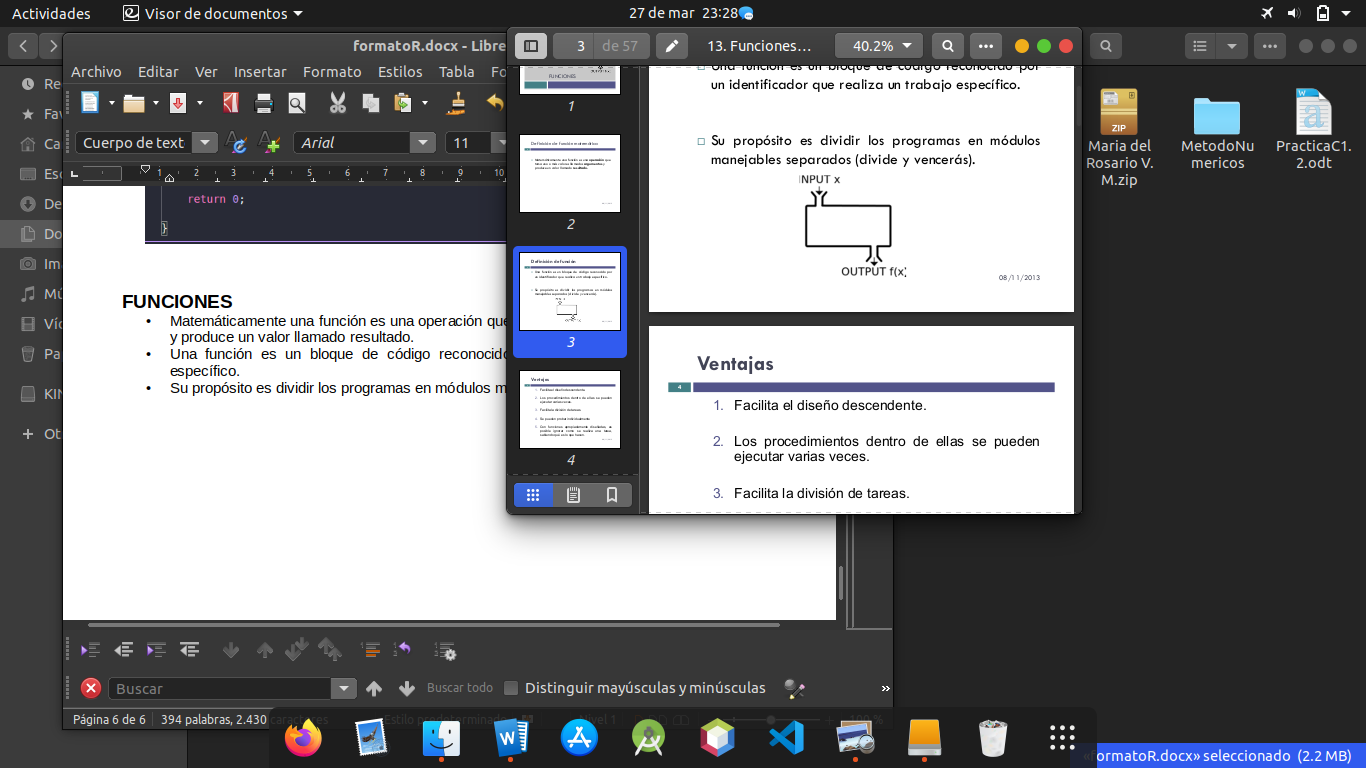
****

**EJERCICIO 1**

****

**FUNCIONES**

* Matemáticamente una función es una operación que toma uno o más valores llamados argumentos y produce un valor llamado resultado.
* 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).



* **Ventajas**

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.

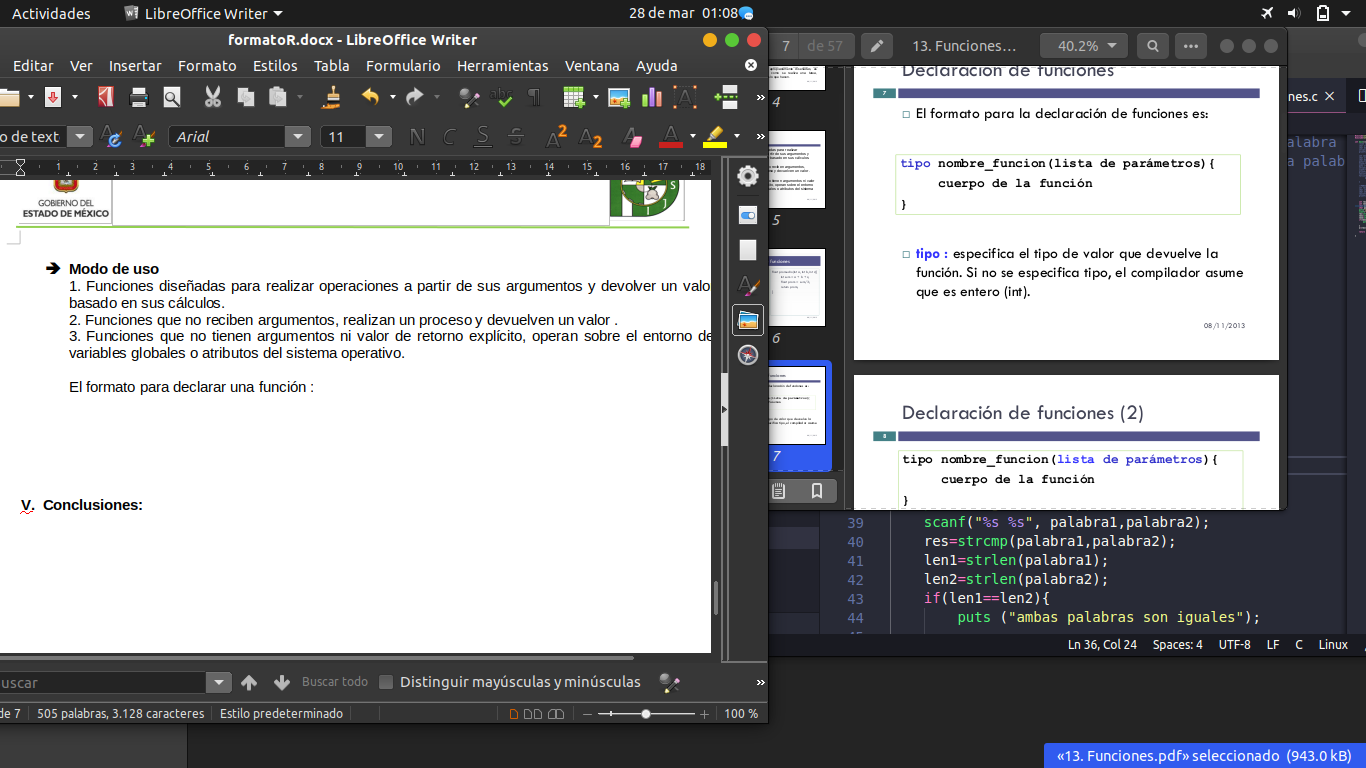
* **Modo de uso**

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

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.

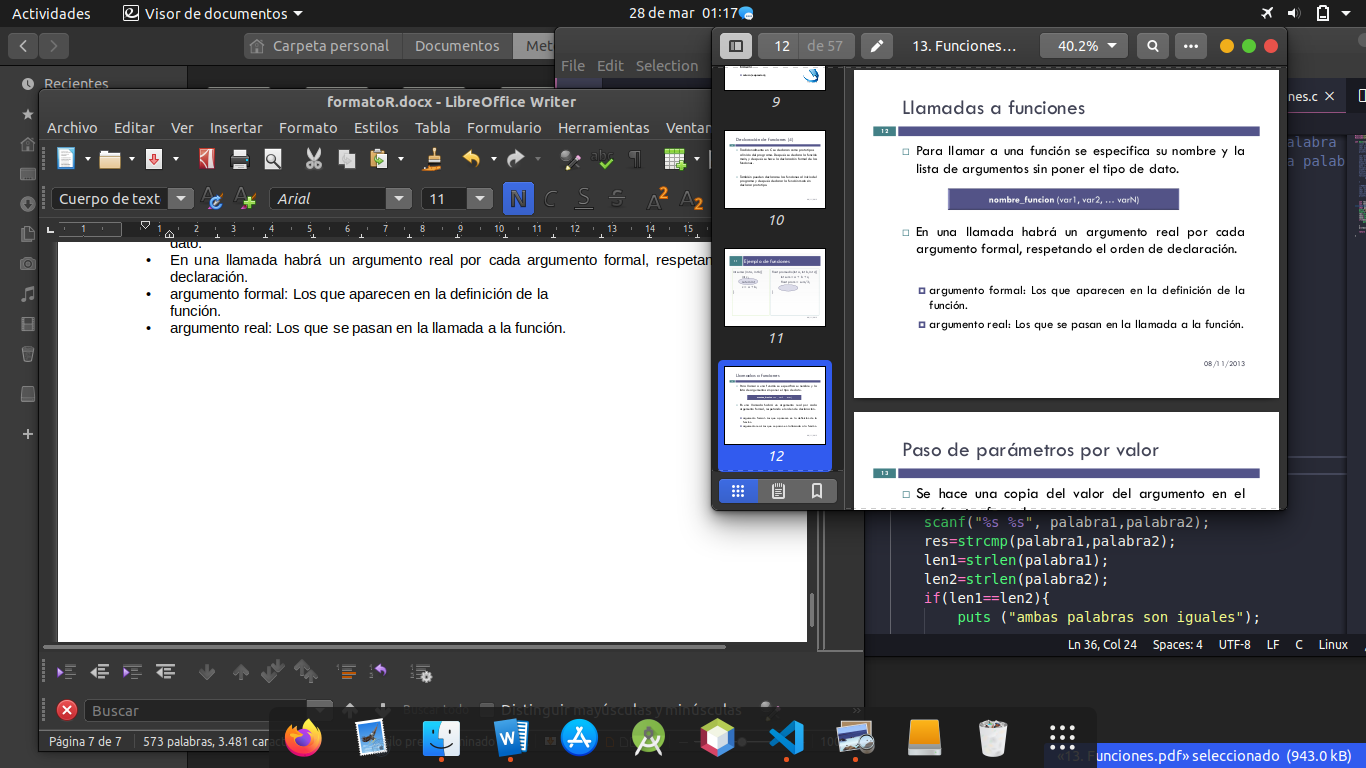
El formato para declarar una función :



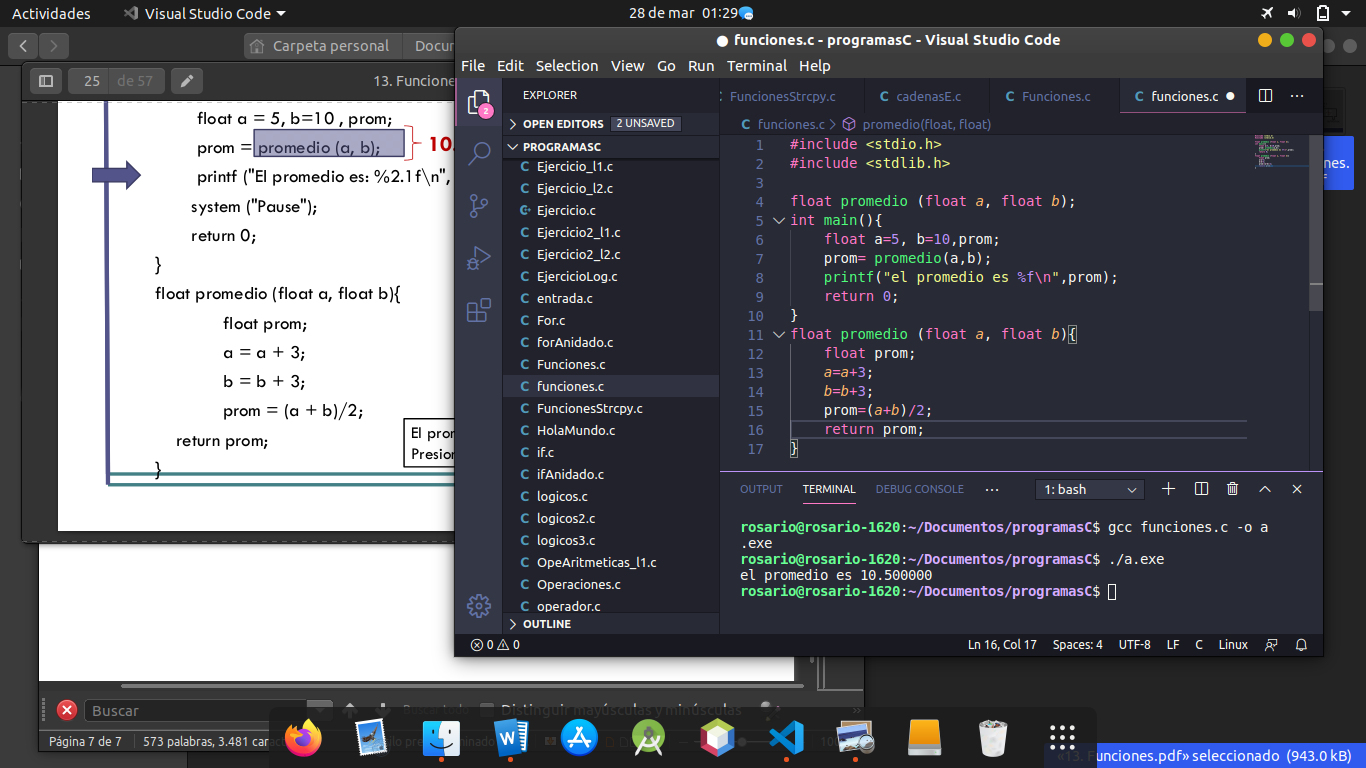
* **Llamadas a funciones**
* Para llamar a una función se especifica su nombre y la lista de argumentos sin poner el tipo de dato.
* En una llamada habrá un argumento real por cada argumento formal, respetando el orden de declaración.
* argumento formal: Los que aparecen en la definición de la

función.

* argumento real: Los que se pasan en la llamada a la función.

****

* **EJEMPLO 1**

****

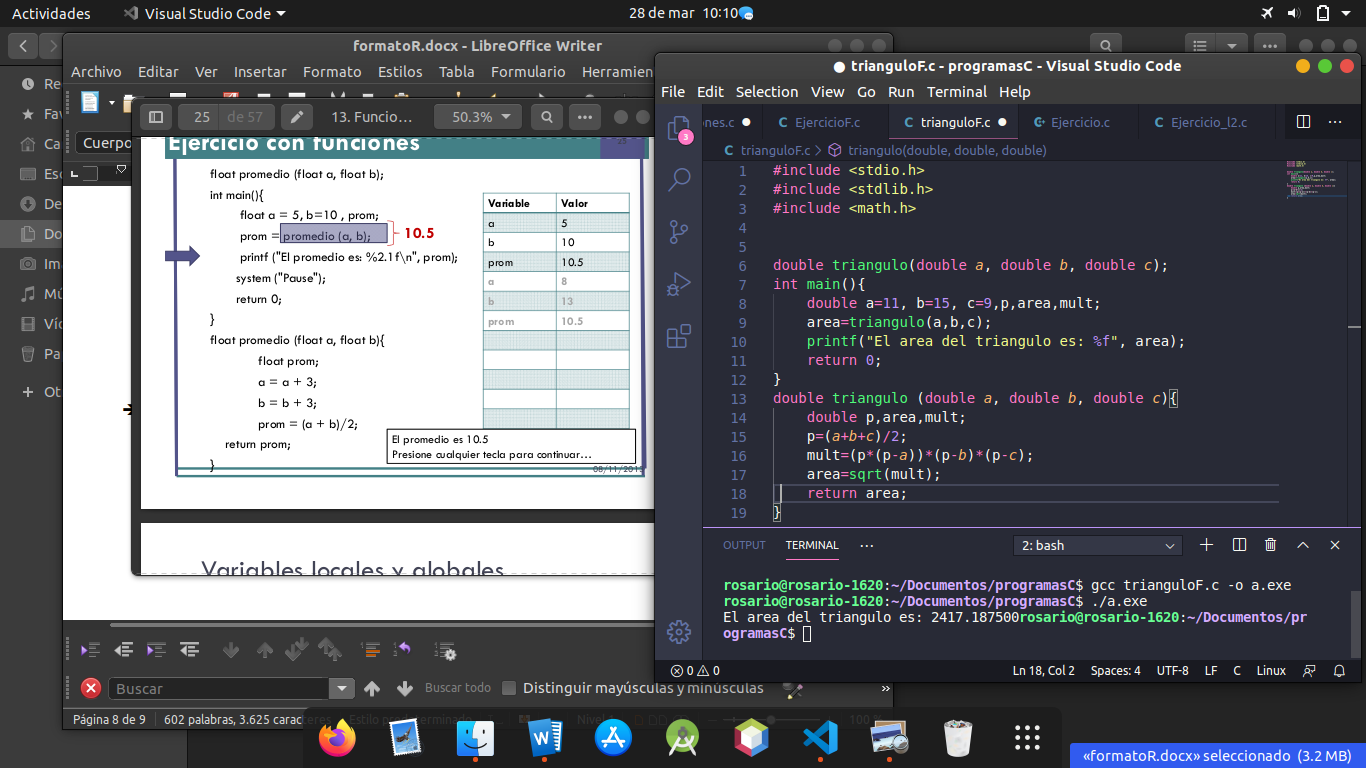
* **EJERCICIO 1**

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

****

* **EJERCICIO 2**

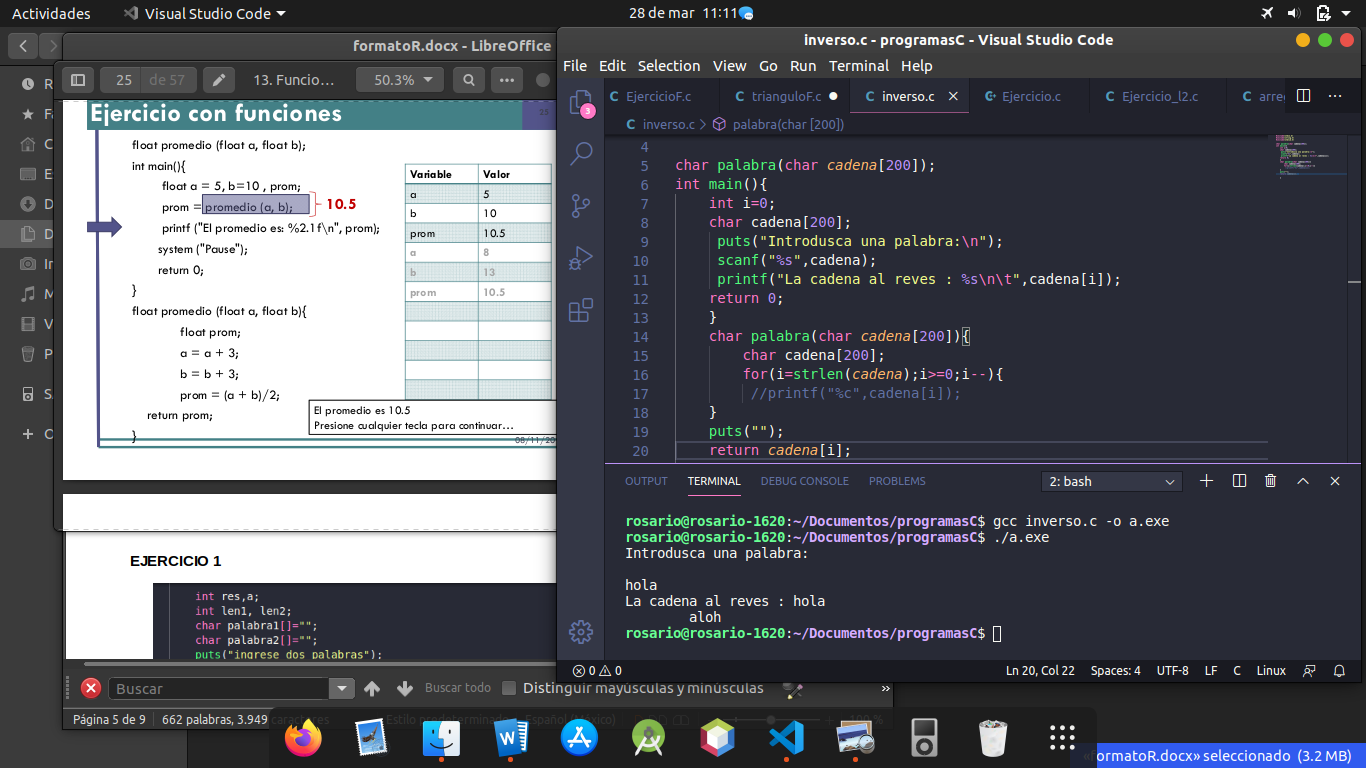
Escribir una función que tome como parámetros las longitudes de los tres lados de un triangulo (a,b, c) y devuelva el área del triangulo.



* **EJERCICIO 3**

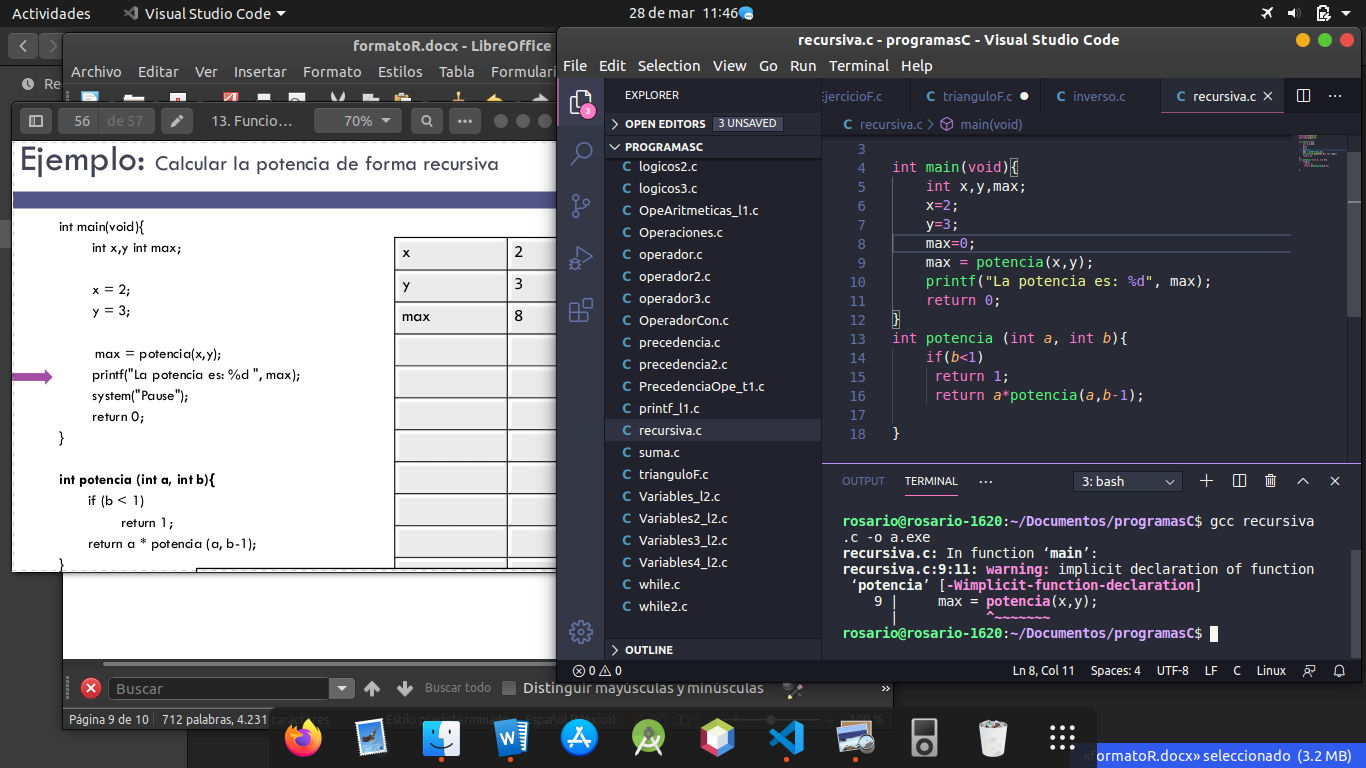
Escribir una función que reciba caracteres del teclado hasta recibir un espacio o un salto de línea

(enter) y a continuación mostrar todos los caracteres en orden inverso.



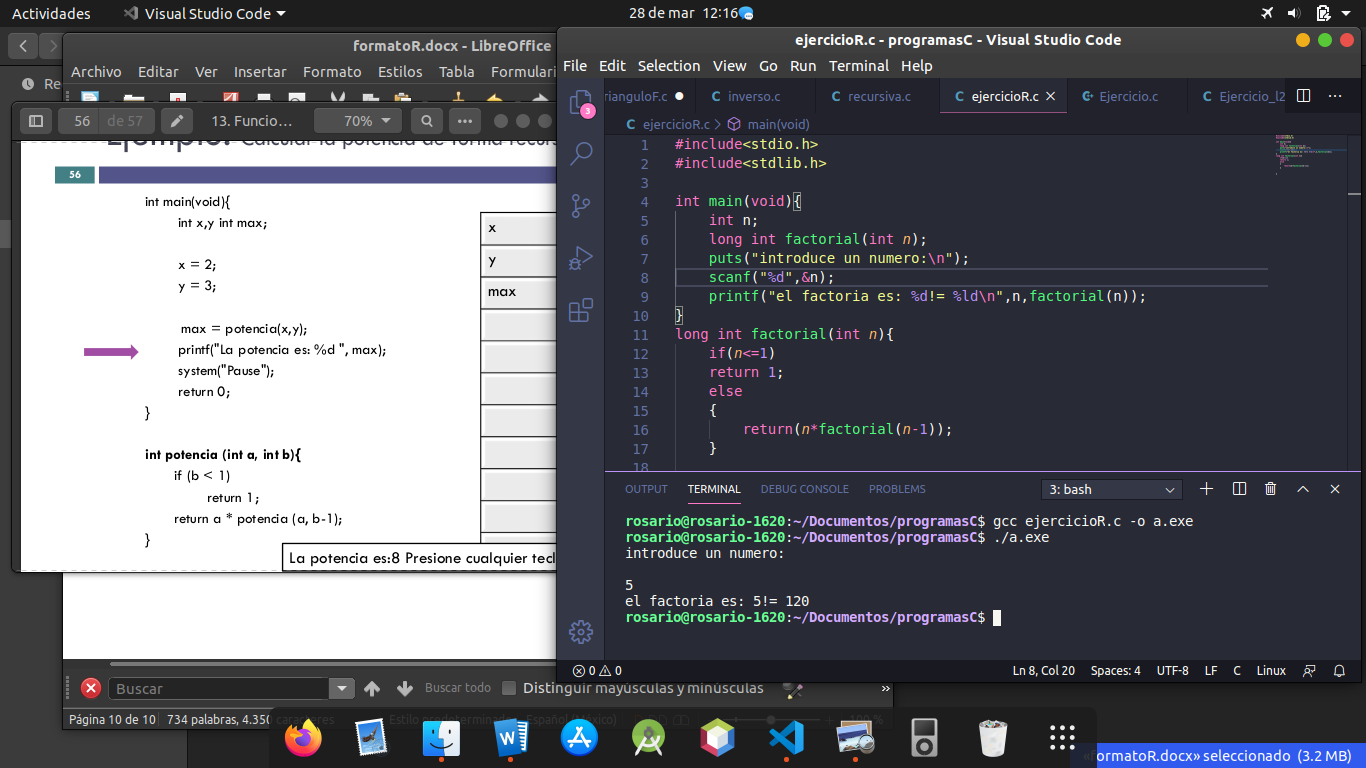
**\***

* **FUNCIONES RECURSIVAS**
* 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
* **EJEMPLO 1** Calcular la potencia de forma recursiva.



* **EJERCICIO 1**

Haz un programa con funciones recursivas que calcule el factorial de un número **n** ingresado desde teclado.

****

**V. Conclusiones:** las cadenas se utilizaran cuando utilicemos caracteres van a permitir concatenar mas caracteres las cadenas tienen diferentes funciones que se ocuparan dependiendo el contexto donde se maneje el problema. Son un tipo de vectores por lo mismo se declara en un arreglo, con vector char.

Las funciones reducen código, haciendo la solucion por separado para solo al final hacer una conexión. Existe variables locales que se utilizan unicamente en una función declarada y variables globales se crean durante toda la ejecución del programa.