|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **NOMBRE DE LA PRÁCTICA** | **Prácticas de C ++** | | | **No.** | | **3** | |
| **ASIGNATURA:** | **Métodos númericos** | **CARRERA:** | **INGENIERIA EN SISTEMAS COMPUTACIONALES** | | **PLAN:** | | **ISIC-2010-224** |

**I. COMPETENCIA(S) ESPECÍFICA(S): Aprender c++**

**II. MATERIAL EMPLEADO:**

* Laptop con Linix Lite
* Editor de texto Visual Studio Code
* Terminal

**III.**

**DESARROLLO DE LA PRÁCTICA:**

Ejemplo 1 de cadenas de caracteres.

* Importaremos las bibliotecas necesarias para el funcionamiento del programa que son el estandar de entrada y salida y la biblioteca para Strings o cadenas de caracteres



* Declarar el método principal



* Declarar la cadena de caracteres cad con espacio para 10 caracteres



* Después mostrar un mensaje al usuario que introduzca la palabra y dar un salto de linea



* Luego guardar lo que ingreso el usuario a el arreglo de caracteres cad



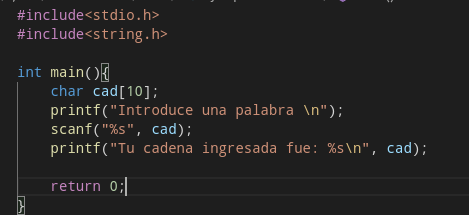
* Después somo mostrar la cadena ingresada al usuario y un salto de linea



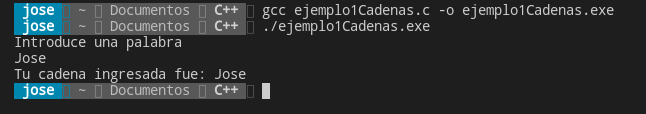
* Retornar 0 y cerrar el método principal



* Código completo:



* Al guardar nuestro archivo como ejemplo1Cadenas con extención c lo siguiente sera compilar y ejecutar nuestro .exe



Ejemplo 2: Longitud de la cadena

* Importaremos las bibliotecas necesarias para el funcionamiento del programa que son el estandar de entrada y salida y la biblioteca para Strings o cadenas de caracteres



* Declarar el método principal



* Declarar una variable de tipo entero que guardara la longitud de nuestra cadena



* Ademas declarar una cadena de caracteres que almacenara el texto “Es una cadena”



* La función strlen nos devolverá el tamaño de la cadena que le pasemos como parámetro y la almacenara en len



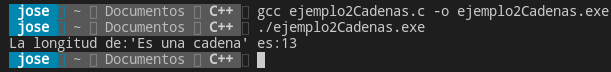
* Al final mostraremos al usuario la cadena y de cuanto es longitud con salto de linea



* Retornar 0 y cerrar el método principal



* Al guardar nuestro archivo como ejemplo2Cadenas con extención c lo siguiente sera compilar y ejecutar nuestro .exe



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).

* Importaremos las bibliotecas necesarias para el funcionamiento del programa que son el estandar de entrada y salida y la biblioteca para Strings o cadenas de caracteres



* Declarar el método principal



* Declarar la cadena de caracteres cadena con espacio para 100 caracteres



* Y tres variables i, j y longitud: las dos primeras serán contadores de los dos ciclos y la última será para almacenar la longitud de cadena



* Primero imprimir un mensaje para decirle al usuario que ingrese una palabra luego guardar lo ingresado por teclado en cadena

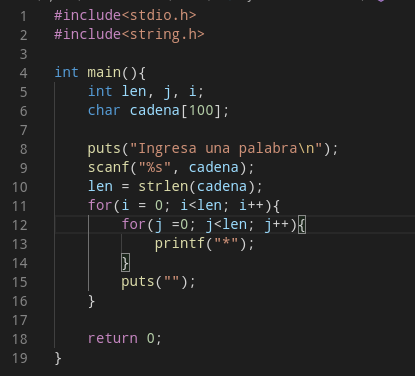


* Después sacar la longitud de la cadena con la función strlen y guardarlo en len

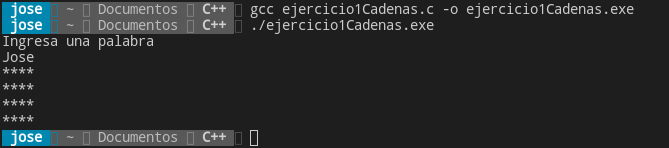


* La lógica del programa esta en las lineas siguientes que son dos ciclos anidados que el primero dara un salto de linea y el anidado imprimirá asteriscos hasta ser igual a la longitud de cadena

Código completo



* Al guardar nuestro archivo como ejercicio1Cadenas con extención c lo siguiente sera compilar y ejecutar nuestro .exe e ingresar por teclado Jose vemos que cumple con lo que pide el programa



**Strcpy**

* Importaremos las bibliotecas necesarias para el funcionamiento del programa que son el estandar de entrada y salida y la biblioteca para Strings o cadenas de caracteres



* Declarar el método principal



* Primero declararemos la cadena de caracteres origen con contenido “Origen”



* Luego otra cadena de caracteres que tendrá 7 espacios en memoria



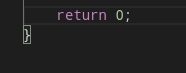
* Después con el método strcpy copiaremos la cadena origen en la cadena destino



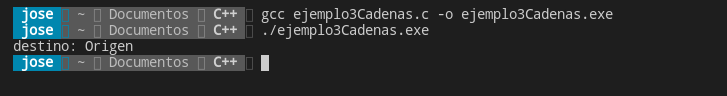
* Para comprobar lo antes dicho se imprimirá la cadena destino con salto de linea



* Por último retornaremos 0 y cerraremos el método principal



* Al guardar nuestro archivo como ejemplo3Cadenas con extención c lo siguiente sera compilar y ejecutar nuestro .exe



Ejercicio 2

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

antes

Palabra\_1 = ' Programacíon'

Palabra \_2= 'Computadora'

después

Palabra\_1 = 'Computadora'

Palabra \_2= 'Programacíon'

* Importaremos las bibliotecas necesarias para el funcionamiento del programa que son el estandar de entrada y salida y la biblioteca para Strings o cadenas de caracteres



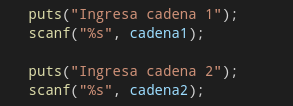
* Declarar el método principal



* Declarare 3 cadenas las dos primeras serán donde se guardaran las dos palabras ingresadas por teclado y la tercera será para hacer un respaldo de la palabra copiada



* Después pediremos las dos cadenas al usuario

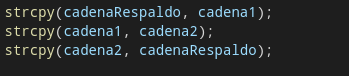


* Aqui va la primera parte del programa donde imprimiremos las cadenas originales



* Esta es la parte más importante del código ya que se divide en pasos

1. Respaldar el valor de la cadena 1 en la cadenaRespaldo
2. Copiar la cadena2 en la cadena1
3. Copiar en la cadena2 el valor de la cadenaRespaldo que es el valor de la primera



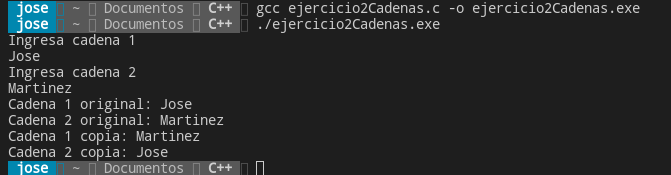
* Después solo quedara imprimir el resultado de los cambios



* Ya por último retornar 0 y cerrar el método principal



* Al guardar nuestro archivo como ejercicio2Cadenas con extención c lo siguiente sera compilar y ejecutar nuestro .exe



**Strcat**

* Importaremos las bibliotecas necesarias para el funcionamiento del programa que son el estandar de entrada y salida y la biblioteca para Strings o cadenas de caracteres



* Declarar el método principal



* Serán declarados dos cadenas que serán origen y destino con contenido “brisas” y “para”



* Con la función strcat funcionara como concatenador de la cadena origen y destino



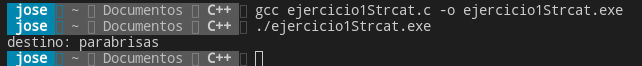
* Después se imprimirá la concatenación de estas dos cadenas



* Y por último se retornara 0 y se cerrara el método principal



Al guardar nuestro archivo como ejemplo1Strcat con extención c lo siguiente sera compilar y ejecutar nuestro .exe



**Ejercicio 3**

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.

Ejemplo:

Palabra1: “para”

Palabra2: “brisas”

Palabra1: parabrisasbrisasbrisasbrisas

* Importaremos las bibliotecas necesarias para el funcionamiento del programa que son el estandar de entrada y salida y la biblioteca para Strings o cadenas de caracteres



* Declarar el método principal



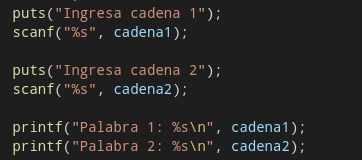
* Declarar dos cadenas de caracteres con nombres cadena1 y cadena2



* Declarar un entero llamado len



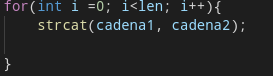
* Pedir al usuario las dos cadenas y luego imprimirlas



* Obtener el tamaño de la cadena1 y almacenarlo en len



* Aquí va la parte más importante del código que es en un for que se repetirá la longitud de la cadena 1 y dentro del for hira el strcat que concatenara la cadena2 en la cadena1



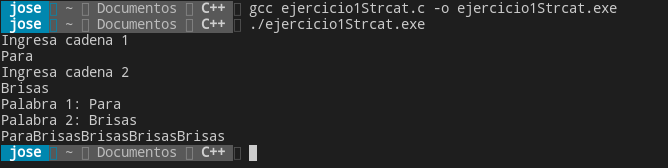
* Después imprimir la cadena 1



* Finalmente retornar 0 y cerrar el método principal



* Al guardar nuestro archivo como ejercicio1Strcat con extención c lo siguiente sera compilar y ejecutar nuestro .exe



**Strcmp**

El strcmp es una función que compara dos cadenas y si las dos cadenas son iguales dará un valor entero de 0 si la primera es mayor dará positivo y si la segunda es mayor dará negativo y su sintaxis es strcmp.

variable= (cadena1, cadena2);

**Ejercicio 4**

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'.

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>.

* Importaremos las bibliotecas necesarias para el funcionamiento del programa que son el estandar de entrada y salida y la biblioteca para Strings o cadenas de caracteres



* Declarar el método principal



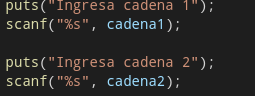
* Declararemos dos cadenas con nombre cadena1 y cadena2



* Declararemos otra variable pero de tipo entero que guardara el valor de la comparación



* Pediremos las cadenas al usuario



* Después va la función de comparación y ya esta resuelto el problema, ya que el valor de la comparación de cadena 1 con la cadena 2 se almacenara en valor



* Imprimimos el valor que almaceno la comparación

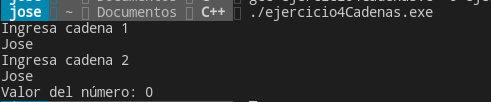


* Por último retornara 0 y cerraremos el método principal

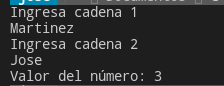


* Al guardar nuestro archivo como ejercicio4Cadenas con extención c lo siguiente sera compilar y ejecutar nuestro .exe

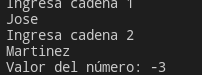
1. Cadenas iguales



1. Cadena 1 mayor



1. Cadena 2 mayor



**Ejercicio 5**

Crea un programa que detecte una palabra palíndroma.

Las palíndromos son palabras que se leen igual de izquierda a derecha que de derecha a izquierda.

Ejemplo: ala, rotor, salas.

* Importaremos las bibliotecas necesarias para el funcionamiento del programa que son el estandar de entrada y salida y la biblioteca para Strings o cadenas de caracteres



* Declarar el método principal



* Declarar dos cadenas 1 que que será donde el usuario ingresara la palabra y otra que será la misma palabra pero inversa para poder hacer la comparación



* Declarar otras dos cadenas que serán el mensaje de la comparación del operador ternario



* Declarar valor que será usado para la comparación, longitud que será el que guarde la longitud de la cadena, i y j que serán contadores



* Pediremos al usuario la palabra o cadena



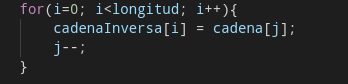
* Para poder hacer el ciclo que invertira la palabra necesitamos saber la longitud de la cadena



* La j será el valor de longitud menos 1 para que abarque la posición correcta



* El ciclo servirá para invertir la palabra de esta manera: i iniciara desde 0 y j iniciara desde la posición más alta haciendo que las dos variables recorran la longitud de la palabra para que cadenaInversa tome el valor inverso de la cadena



* Después valor hará la comparación de las dos cadenas



* Una buena forma de mostrar el resultado será usando el operador ternario que funciona así: Si la valor es igual a 0 quiere decir que es palindroma y se mostrara el mensaje correspondiente, en caso contrario mostrara que no lo es

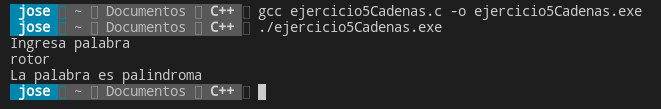


* Por último retornar 0 y cerrar el método principal

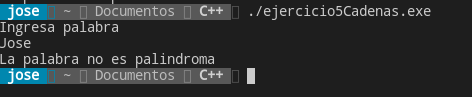


* Al guardar nuestro archivo como ejercicio5Cadenas con extención c lo siguiente sera compilar y ejecutar nuestro .exe

1. Las palabras palindromas:



1. Las palabras no son palindromas



**Ejemplo 6**

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

Ejemplo:

Palabra

P→1

a →3

l →1

b →1

r →1

* Importaremos las bibliotecas necesarias para el funcionamiento del programa que son el estándar de entrada y salida y la biblioteca para Strings o cadenas de caracteres



* Declarar el método principal



* Declaramos una cadena llamada palabra para que aloje la cadena de caracteres



* Lo siguiente es declarar variables de tipo entero las dos primeras serán para los ciclos y la longitud alojara el tamaño de la cadena ingresada



* La siguiente variable de tipo entero llamada contador almacenara temporalmente el número de incidencias en la palabra



* Después se pedirá al usuario la palabra



* Después con la función de strlen obtendremos la longitud de la palabra



* Seguidamente declarar un arreglo que tendrá el tamaño de la palabra



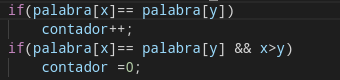
* Después un ciclo que se repetirá de la longitud de la palabra hasta llegar a ser igual a 0



* Un ciclo anidado que será igual al otro



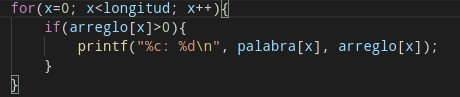
* Después si la posición x de la palabra es igual a la posición y de la palabra se aumentara 1 al contador, después otra condición casi igual pero con la diferencia que ahora si x>y el contador será igual a 0



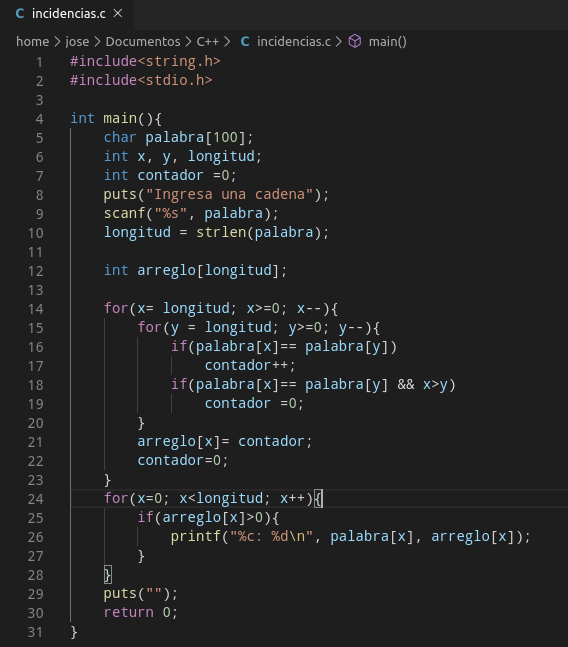
* Luego el arreglo en posición x será igual al contador dependiendo que valor obtuvo este en las condiciones anteriores, y luego el contador se igualará a 0



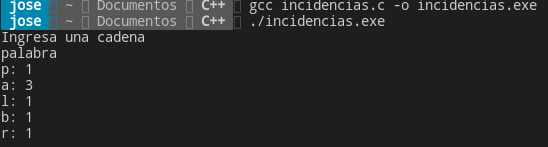
* Después imprimiremos el contador por lo cual lo ingresaremos en un ciclo for, la condición es porque las palabras que incidieron más de una vez el valor del arreglo en esa posición es igual a 0 y asi se hace el filtro de las palabras de la misma manera



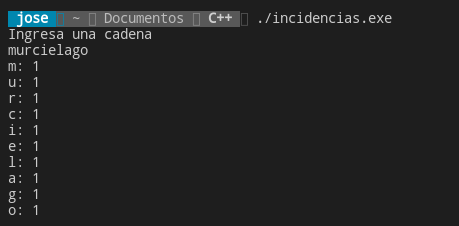
* Código completo



* Al guardar nuestro archivo como incidencias con extención c lo siguiente sera compilar y ejecutar nuestro .exe
* Ingresaremos como cadena de caracteres la palabra: “palabra”



* Ahora intentaremos con la palabra: “murcielago” ya que no tiene ninguna incidencia



**Lámina 13 Funciones**

* Primero importaremos el estándar de entrada y salida



* Después declarar la función promedio con parametros a y b de tipo int



* Declarar el método principal



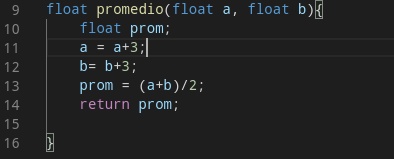
* Luego declarar 2 variables de tipo float que serán los valores que pasaremos a la función y un 3 que será el que almacenara el promedio



* Luego el prom tomará como valor el resultado de la función promedio y le paseremos como parámetros a y b



* La función promedio se le pasan dos parámetros, a los dos se les sumara 3 y el proemdio será la suma de ay b dividido entre 2 y se retornara el prom



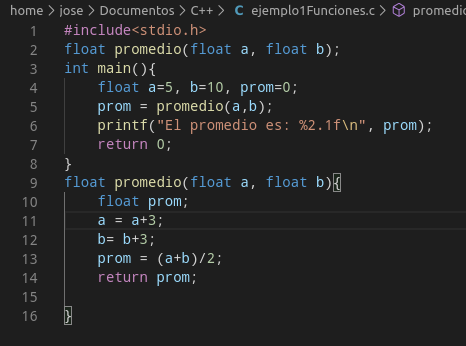
* Se mostrara en pantalla el promedio con dos 1 decima y salto de linea



* Después retornar 0



* Código completo:



Al guardar nuestro archivo como ejemplo1Funciones con extención c lo siguiente sera compilar y ejecutar nuestro .exe

* El resultado es 10.5



**Ejercicio 1 Funciones**

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

* Primero importaremos el estándar de entrada y salida

****

* Declarar la función de tipo int llamada calcularMaximo con dos parámetros



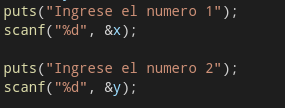
* El método principal sin tipo de retorno



* Declarar 2 variables de tipo entero que almacenaran los números por teclado



* Pedir al usuario que ingrese los número por teclado y almacenarlos en x, y respectivamente



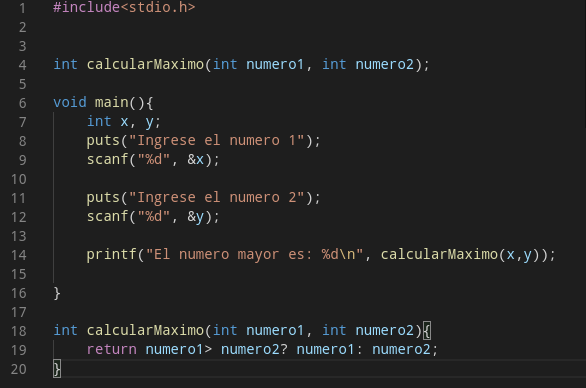
* Imprimir el resultado de nuestra función calcularMaximo pasandole como parametros lo que ingreso el usuario por teclad, y dar un salto de linea



* Para la función solo se recibirán dos parámetros enteros y retornar el número más grande de ambos, aquí use un operador ternario que si el numero1 es mayor que el 2, se retornara el primero y si no el segundo



* Código completo



* Al guardar nuestro archivo como parametros con extención c lo siguiente sera compilar y ejecutar nuestro .exe



**Ejercicio 2**

**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.**

** Ejemplo:**

** Entrada:**

**Hola**

** Salida: aloH**

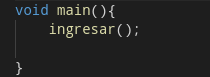
* Importaremos las bibliotecas necesarias para el funcionamiento del programa que son el estándar de entrada y salida y la biblioteca para Strings o cadenas de caracteres



* Después declararemos la función sin parametros y sin tipo de retorno

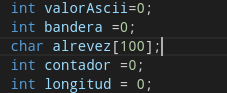


* Después declarar el método principal y mandar a llamar la función



* Después declarar las siguientes variables

1. Almacenara el valor en código Ascii
2. Servirá para detener el ciclo while
3. Almacenara los caracteres de la palabra
4. Se utilizara como contador
5. Guardara el tamaño de la palabra



* Mandar mensaje que ingrese la cadena y declararla



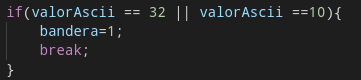
* Iniciaremos un ciclo while que se repetirá hasta que la bandera no sea igual a 0



* Guardar el carácter que ingresa el usuario y guardarlo en cadena, después convertirlo a código Ascii y guardarlo en valorAscii



* La siguiente linea de código es la más importante ya que dice que si el valor ascii es igual a 32-espacio o 10-nueva linea (enter) parara el ciclo while







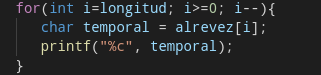
* Luego en el arreglo alrevez en la posición contador será igual a cadena, e incrementar el contador en uno



* Después obtener el tamaño de la cadena con la función strlen y guardarla en longitud



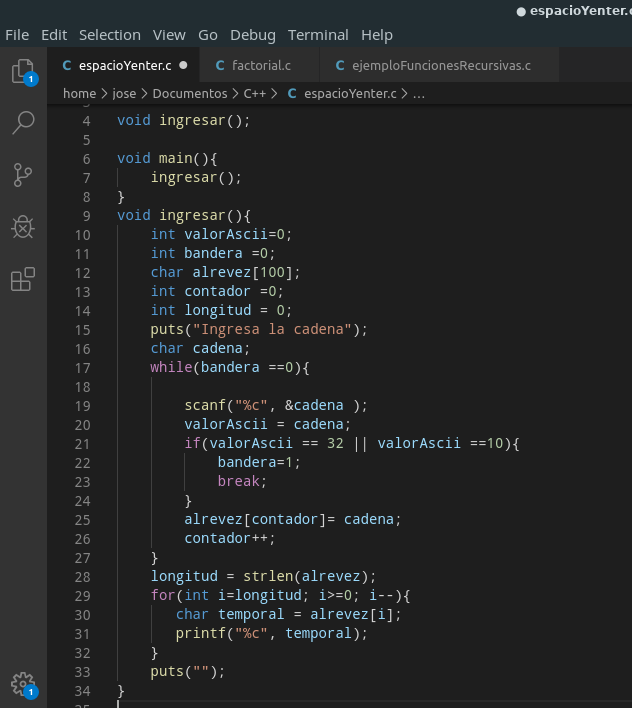
* A continuación imprimir el arreglo al revez en forma inversa



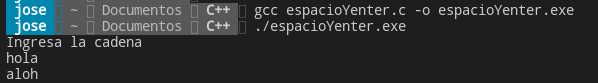
* Dar un salto de linea

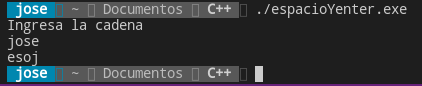


* Código completo



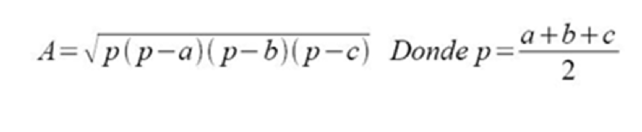
Al guardar nuestro archivo como espacioYenter con extención c lo siguiente sera compilar y ejecutar nuestro .exe



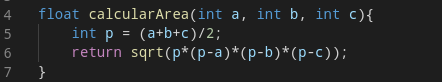


**Ejercicio 3**

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.



* Lo primero será importar 3 biblioteca: 1 el estándar de entrada y salida, 2 el estándar de bibliotecas, 3 la biblioteca para poder hacer la raíz cuadrada
* Después declarar la función de tipo flotante que pedira 3 parametros de tipo entero, declarar una variable llamada p que hará el primer cálculo y se retornara la raiz de p\*(p-a)\*(p-b)\*(p-c)

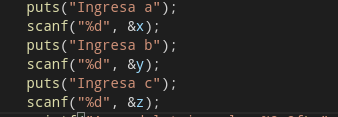


* Declarar nuestro método principal



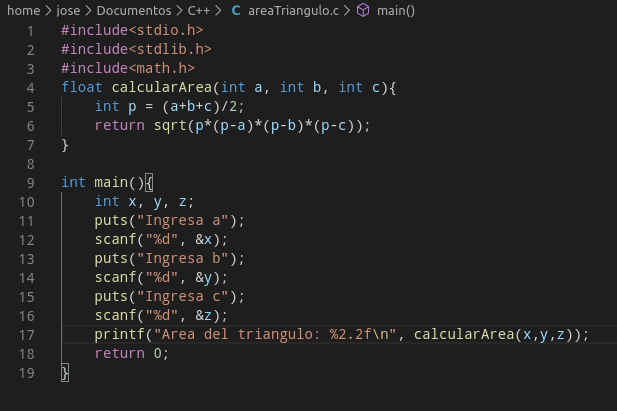
* Declarar 3 variables de tipo entero x, y, z



* Pedir por teclado las variables antes declaradas
* Imprimir el resultado de la operación con 2 decimales, dar salto de linea y retornar



* Código completo:



* Al guardar nuestro archivo como areaTriangulo con extención c lo siguiente sera compilar y ejecutar nuestro .exe
* Aqui debido a que el s.o es linux se agregara -lm al momento de compilarlo

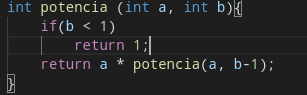


**Ejemplo funciones recursivas**

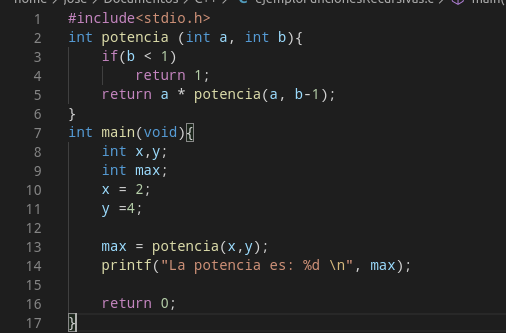
* Importas la biblioteca de estándar de entrada y salida



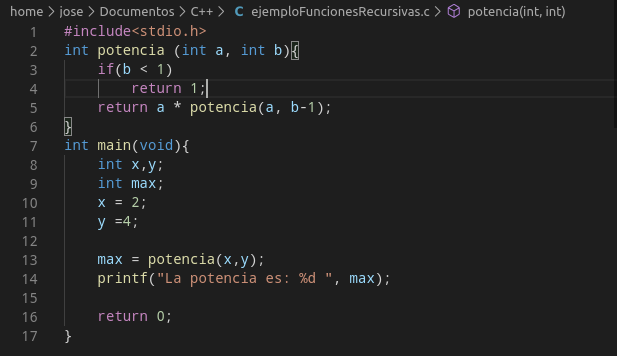
* Después declarar la función potencia que recibirá dos parametros enteros, se repetira hasta que b sea menor a 1 y retornar a\* potencia(a, b-1) para que vaya disminuyendo y se pare al llegar a 0



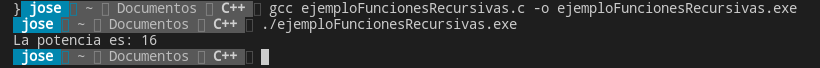
* Luego declarar el método principal, declarar dos variables y darle valor, llamar la función potencia que tomara como parámetros las variables antes declaradas y almacenar su valor en la variable max, por último imprimir la variable max y dar salto de linea y retornar 0



* Código completo

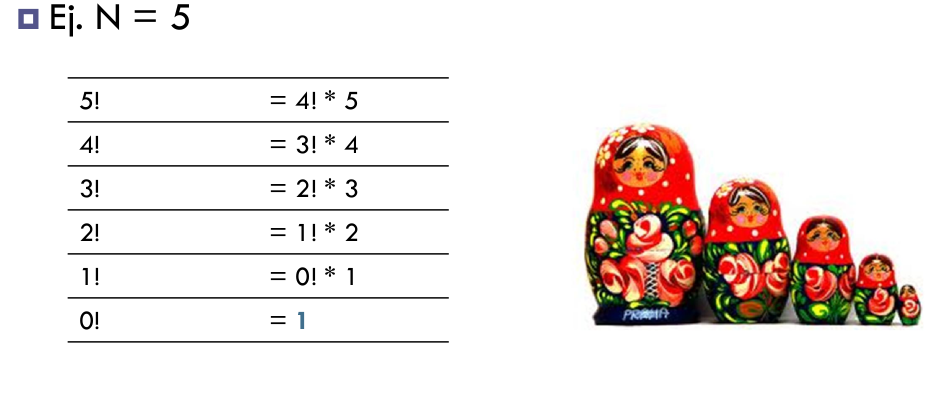


* Al guardar nuestro archivo como ejemploFuncionesRecursivas con extención c lo siguiente sera compilar y ejecutar nuestro .exe



**Ejercicio Funciones Recursivas**

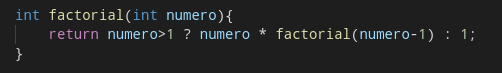
Haz un programa con funciones recursivas que calcule el factorial de un número n ingresado desde teclado.

****

* Lo primero es importar la biblioteca de estándar de entrada y salida



* Después declarar la función llamada factorial que recibirá un parámetro llamado número y retornara un número entero.
* Esto lo hize con un operador ternario que si el numero es mayor 1 retornara el número multiplicado por el número menos 1 y si no retornar 1 que es el factorial de 0



* Declarar el método principal



* Declarar una variable de tipo int que se llamara n



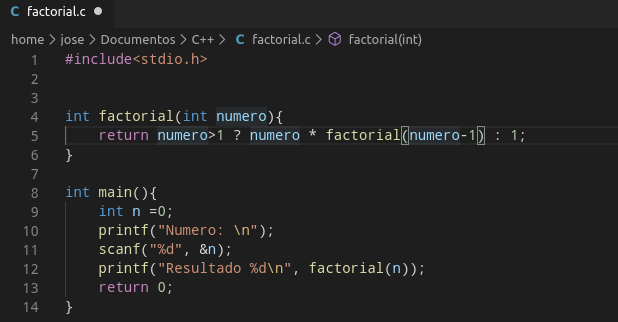
* Pedir por teclado el número y almacenarlo en la variable n



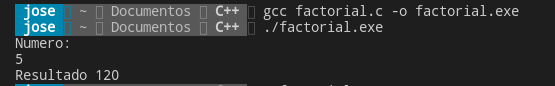
* Lo último será imprimir el resultado llamando la función pasando le como parámetro n, dando salto de linea y retornando 0



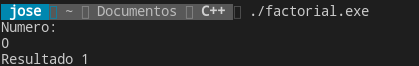
* Código completo:



* Al guardar nuestro archivo como factorial con extención c lo siguiente sera compilar y ejecutar nuestro .exe
* El ejemplo propuesto



* El número igual a 0



**IV. Conclusión:**

Una vez realizado los problemas y ejemplos propuestos llego a la conclusión que esto de las cadenas si se ocupa en gran magnitud ya que casi todos los programas las usamos y manipularla se vuelve indispensable, las funciones también no se quedan atrás ya que como dijo el profesor: divide y vencerás y mucho más las funciones recursivas son de gran ayuda a la hora de programar correctamente