|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **NOMBRE DE LA PRÁCTICA** | **Cadena** | | | **No.** | **1** |
| **ASIGNATURA:** | **METODOS NUMÉRICOS** | **CARRERA:** | **INGENIERIA EN SISTEMAS COMPUTACIONALES** | **PLAN:** | **ISIC-2010-224** |

**VERONICA CRUZOSORIO**

1. **COMPETENCIA(S) ESPECÍFICA(S):**

**II. MATERIAL EMPLEADO:**

Dev-c++

COMPUTADORA

**III.DESARROLLO DE LA PRÁCTICA:**

**¿Qué es una cadena?**

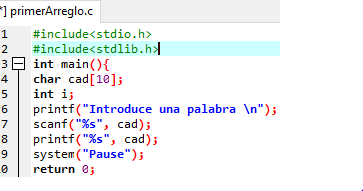
Una cadena es un arreglo de caracteres. En donde, por lo general el último elemento deberá ser el carácter ‘\0’.

**¿Cómo declarar una cadena?**

Crearla como un arreglo **char c [10];**

Bibliotecas

Se crea un arreglo



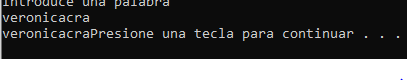
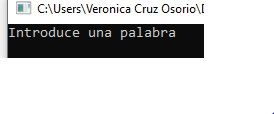
Y es mostrado en pantalla

Con **printf** nos ayuda a mandar un mensaje en pantalla y los datos que ingresen por teclado y este es asignado a la variable **cad**

Método **main**.

Se crea un arreglo de ti **char** y que lleva de nombre **cad** y que tiene un tamaño 10 lugares, pero recordemos que debemos de contar desde 0

**¿**

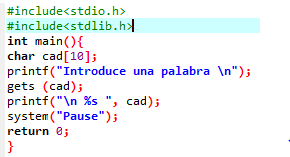


Bibliotecas

**Ejemplo de una cadena**

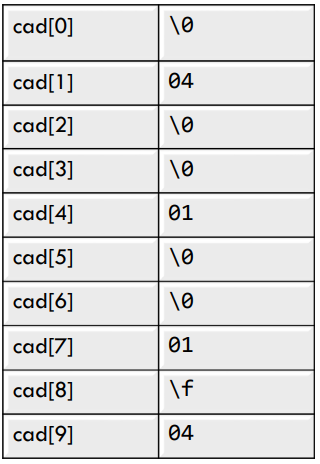
Método **main,**

Declara un arreglo de un tamaño de 10 con nombre **cad**



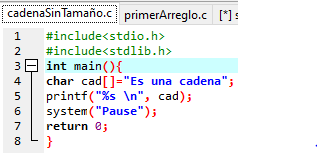
Se manda un mensaje en pantalla para que el usuario ingrese una palabra y e gets es

es leer todo lo que haya en el buffer hasta que encuentre un '\n'



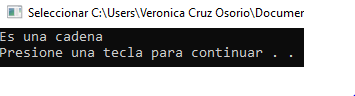
**Ejemplo de una cadena sin tamaño**

Crearla como arreglo sin tamaño ν char cad [] = “Es una cadena”;



Se declara una cadena sin tamaño pero es igual a **“Es una cadena”,** se manda un mensaje en pantalla con el valor de la cadena.

Bibliotecas

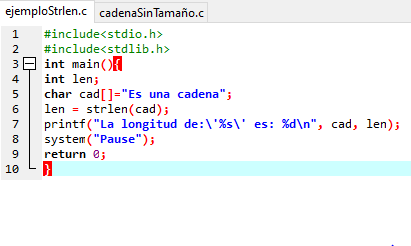


El mensaje que fue mandado en pantalla. Y que tiene el valor de la cadena

**Funciones de cadena: strlen**

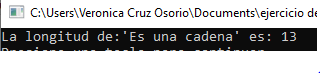
Devuelve la longitud de la cadena sin tomar en cuenta el carácter de final de cadena. **strlen().**

Bibliotecas



Calcula el número de caracteres de la cadena, el valor que fue asignado a la cadena **cad y**  se manda a imprimir el valor de **cad** que es “Es una cadena” y su longitud que es 13 contando con espacions.

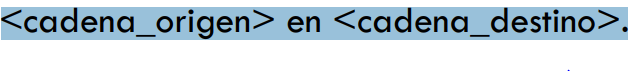
Se declara una variable de tipo entero con nombre **len,** despues se declara una cadena sin tamaño con nombre **cad** pero es igual a “Es una cadena”y este valor se le asigna a la variable **len**



Se muestra en pantalla el mensaje + el valor que tiene **cad** + la longitud (13)

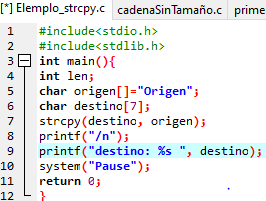
**Funciones de cadena: strcpy**

Copia el contenido de: en . strcpy(, )





Bibliotecas

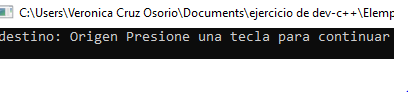


Sirve para copiar la cadena cadena2 dentro de cadena1. Devuelve el valor de cadena1. strcpy recibe dos parámetros: un apuntador a char (modificable), y un apuntador a char constante. Y devuelve un apuntador a char

Se declara una variable de tipo entero de nombre **len .**  se declara una cadena 1 de tipo char de nombre **origen** que no tiene ningún tamaño pero es igual de la cadena 2 **“origen”,**  y es declarado otra variable del mismo tip y de nombre destino pero este tiene un tamaño de 7.

Con ayuda de **strcpy**  en este caso nos ayuda a que la cadena 2 dentro de la cadena 1 y se muestre un mensaje donde esta destino: y devuelve el valor de la cadena 1 que es **origen**  pero se manda a llamar para imprimir la cadena destino porque está dentro de la cadena 1.

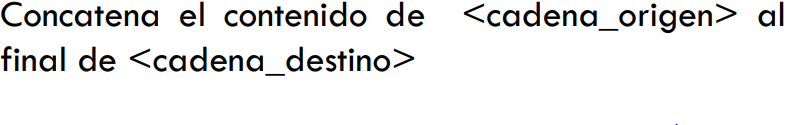
Se muestra en pantalla el mensaje de **destino: Origen**



**Ejercicio. Escribe un programa donde se ingresen dos palabras por teclado y al imprimir la primera en la posición de la segundo y la segunda en la posicione la primera.**

**Funciones de cadena: strcat**

**Sirve para concatenar la cadena cadena2 hacia cadena1. Devuelve el valor de cadena1.**

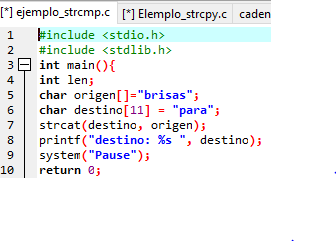




Escribe un programa que reciba por teclado dos palabras

Bibliotecas

s

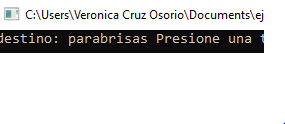


Se manda a imprimir el mesaje donde destino tiene la palabra **para** + destino que es donde se guarda la palabra **brisas**

Dspues se usa strcat Sirve para concatenar la cadena cadena2 hacia cadena1. Devuelve el valor de cadena1. Y se manda concatenar la cadena 2 dentro de la 1

Se declara una variable de tipo entero de nombre **len,** se declara dos cadenas de tipo char de nombres origen y destino, en la primera cadena que es origen no tiene tamaño pero es igual a **“brisas”**  y la cadena 2 si tiene tamaño de 11 pero también es igual a **“para”**

Se manda a imprimir la palabra **parabrisas**

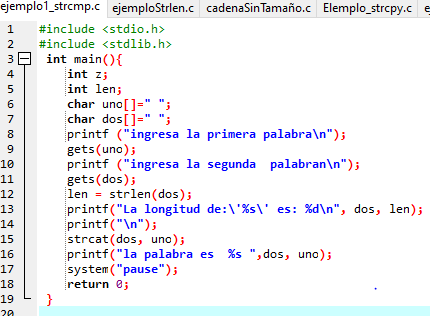


**Ejemplo**

Y concatene N veces la segunda palabra a la primer palabra.

Donde N es la longitud de la primera palabra.

Bibliotecas

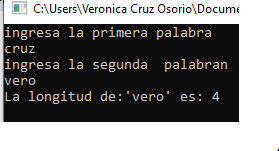


Se manda a imprimir la cadena dos que tiene la palabra **para**  y la uno que es **brisas**

Vamos a mostrar la longitud de la palabra 2 serán las veces que se va a concatenar la palabra y mostrar en pantalla.

Se manda un mensaje para que se ingrese la palabra uno y dos y se le asigne a la cadena 1 y 2

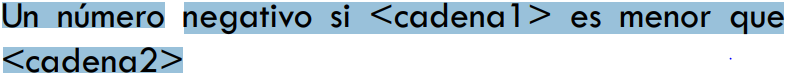
Se declaran cadenas de tipo **char** pero no tiene ningún tamaño

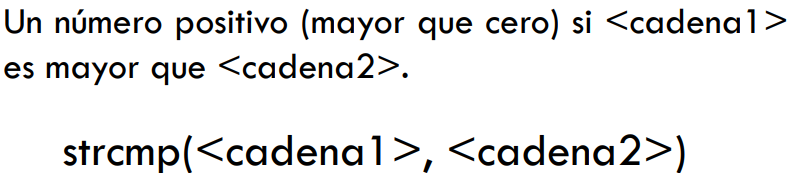


Se manda a imprimir en pantalla la palabra concatenada

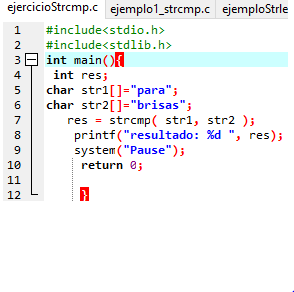
**Funciones de cadena: strcmp**

Compara las dos cadenas y devuelve un 0 si las dos cadenas son iguales.





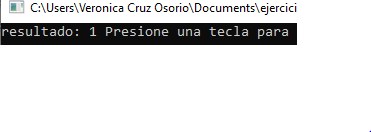
Bibliotecas



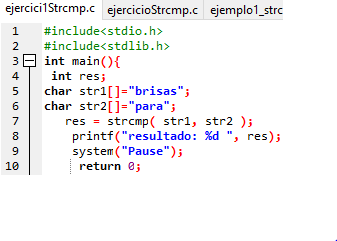
En la variable **res** en seguida **strcmp**  se compara cuál de las dos palabras son más pequeña y como el primer cadena tiene un tamaño de 4 y la 2 tiene un tamaño de 6 entonces la más pequeña es la 1 y ese valor se queda en **res**  y se manda a imprimir en pantalla.

Se declara una variable de tipo entero y de nombre **res.**

Se declara una cadena de tipo char con nombre str1 que es igual **“para”** y str2  que es igual **“brisas”.**



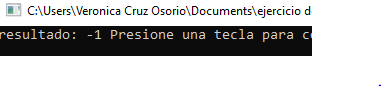
Se muestra en pantalla el resultado que es 1 la palabra menor.

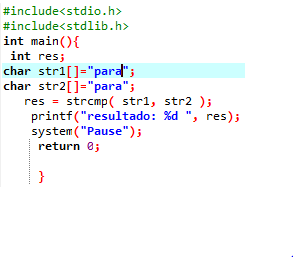


Se muestra el resultado en pantalla

Es parecido al anterior pero este se cambia las palabras donde la cadena 1 es igual a brisas y la 2 es igual a para.

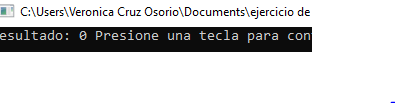
El resultado es igual a -1





Devuelve el valor 0 si ambas cadenas tienen el mismo valor, tamaño

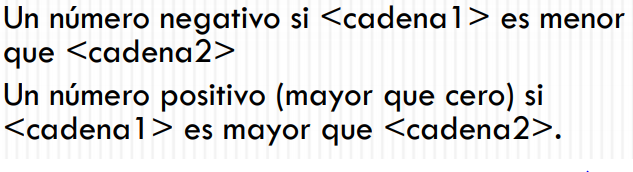
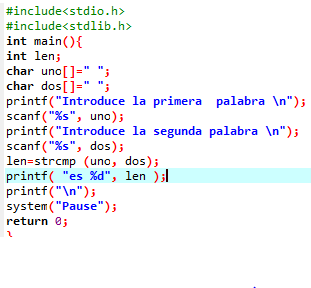
Se cambian las cadenas y ambas son iguales a **“para”** y el resultado es 0



Muestra en pantalla el valor 0 indicando que ambas son iguales

**Ejemplo**

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'.



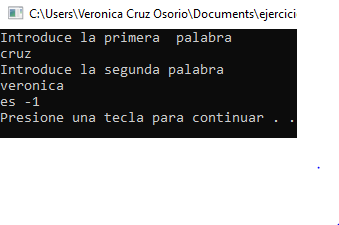
Biblioteca

De clara dos cadenas de tipo **char** de nombre uno y dos que no tiene tamaño.

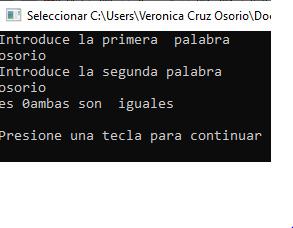
Se muestra un mensaje para que el usuario introduzca la primera palabra que esta se almacenara en la cadena uno y la siguiente e n la dos,

Entonces al inicio se declaró una variable de tipo entero **len**  ella almacenara cuál de las dos palabras es mayor o menor o igual que la otra

Se manda a imprimir en pantalla **1** si la cadena 1 es mayo que la cadena 2, -**1** si la cadena 1 es menor que la cadena 2, **0**  si ambas cadenas son iguales

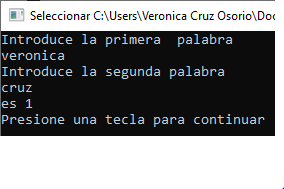


La primera cadena es menor que la segunda entonces nos da un valor de -1



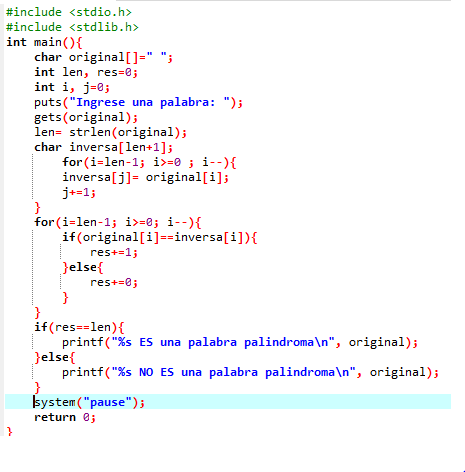
Ambas palabras son iguales entonces nos imprime un mensaje “ambas son iguales ”

Nos manda un 1 si la primera cadena es menos que la segunda



**Ejemplo**

♦ Crea un programa que detecte una palabra palíndroma. ⁄ Los palíndromos son palabras que se leen igual de izquierda a derecha que de derecha a izquierda. ⁄ Ejemplo: ala, rotor, salas.



Se nada un mensaje donde se nuestra si la palabra es palíndroma o no.

Donde cada uno de los for va a leer la palabra de derecha a izquierda para saber si ambos lados tienen lo mismo como puntero, y asi se pueda identificar si la palabra se lee del mismo lado como del otra

Un cilo **for**  donde i= a len meno 1, 1 es menor que 0 y i de decrementa..

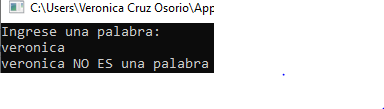
Y donde la cadena inversa tiene el tamaño de j es igual a la cadena original que tiene el valor de la variable i

Se declara una cadena de tipo **char** con nombre original pero no tiene tamaño. Se declaran 2 variables de tipo entero con nombre **len, i, y res=0 &j=0.**  Se manda un mensaje con la ayuda de un puts que ingresar una palabra, y el dato que ingrese se va almacenar en la variable original.

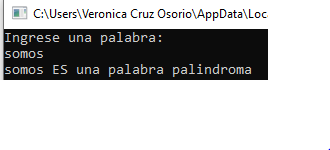
Se le va a asignar el valor de original a **len**

Otra cadena de tipo char con nombre inversa y tiene el mismo tamaño que la variable **len** +1.

Bibliotecas



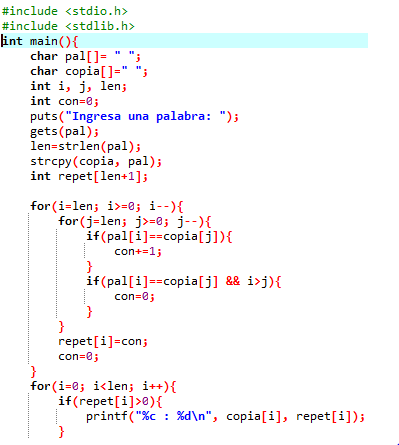
Se muestra en pantalla el texto y se ingresa una palabra nuestra que no es una palabra palindroma



Y la otra palabra se lee de ambos lados y es correcto si es palindroma

**Ejemplo**

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



Donde if evalúa la cadena pal es igual a la copia y menor que j

Donde j tiene el mismo valor que len y donde la condición es que j es meno r que 0 y j se degremanta, y un un ciclo if ayuda a evaluar si es falso o verdadero a cadena pal tiene el tamaño de i y es igual igual a u la cadena copia que tiene el valor que j.

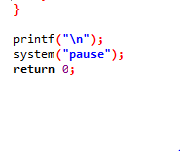
Se declara 2 cadenas de tipo char con nombre pal y copia pero no tiene tamaño, y 3 variables de tipo entero y una de nombre con=0,

Se manda un mensaje con ayuda de **puts**  donde se debe ingresar una palabra, y este valor se asigna ala variable pal, donde la variable len se le signa lo que tiene la variable pal; strlen,

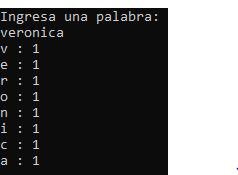
Strcpy se copia un valor de una variable copia y pal.

Y donde se declara una cadena de tipo entero de nombre repet y que tiene un tamaño de len +1.

Biblioteca



Y se evaua que letras se repiten, pasando de la cadena de ambos lados para que una letra se toma y se evalue con el resto de las celdas y si si se valla almacenando en un acomulador.



Se muestra en pantalla para ingresar una palabra y se evalúa si una letra de la palabra se repite con el resto de la cadena

**Funciones**

**¿Qué es una función?**

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

**Hacerlo por modulos**

**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

Diseño descendente, división de tareas,

**Modo de uso?**

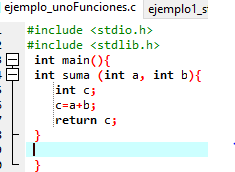
Funciones diseñadas para realizar operaciones a partir de sus argumentos y devolver un valor basado en sus cálculos.

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 atributos del sistema operativo.

**Ejemplo de función**

Bibliotecas



Tenemos un módulo que se llama suma y tiene dos variables **a y b** que son de tipo entero y se declara otra variable **c** de tipo entero  y se ejecuta la operación que **c=a+b** y retorna en **c**  por que es de tipo entero.

Método main

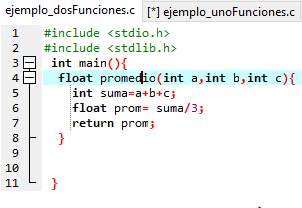
Bibliotecas

**Ejemplo de función 2**

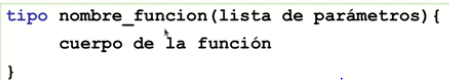
Tenemos un módulo que se llama **promedio** y tiene tres variables **a, b** y **c** que son de tipo entero.

Después esta la variable suma de tipo entero y que obtener el valor de ella es necesario que se ejecute la operación que tiene que es la suma de las tres variables y después una variable de tipo **float** (valores con punto decimal) que es **prom** que es el promedio y que se tiene que ejecutar la operación que tiene dentro que suma (el resultado de la operación de suma ) /3.

Después el retorno en **prom** que es de tipo flotante



**Como se declara una función.**



Como se llamara la función. Y **lista de parámetros**: es la lista de nombres de variables separados por comas con sus tipos asociados que reciben los valores de los argumentos actuales de la llamada a la función.

El tipo de dato tipo : especifica el tipo de valor que devuelve la función. Si no se especifica tipo, el compilador asume que es entero (int).

**Para llamar a una función**

Se debe de respetar el orden de que sean declarado los argumentos

El nombre de la función y la lista de argumentos sin poner el tipo de dato.



Habrá un argumento real por cada argumento formal, respetando el orden de declaración

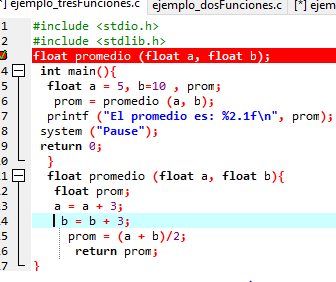
**Pasos de los parámetros por valor**

* Se hace una copia del valor del argumento en el parámetro formal.
* La función opera internamente con estos últimos.
* Los parámetros formales se crean al entrar a la función y se destruyen al salir de ella, cualquier cambio realizado por la función en los parámetros formales no tienen ningún efecto sobre los argumentos.

**Ejemplos de ejercicios de programas**

El nombre de la función y su tipo; en este caso es de tipo **float**  y se llama  **promedio**, con dos argumentos de tipo float **a** & **b.**

Bibliotecas

**=**

Donde **a** es un acomulador; y se realiza su operación; a=5

a=5+3=8 & b=10+3=13

y el **prom**=el resultado de la operación de **a + b** y se suma **8+13=21** y el resultado **21/ 2** y =10.5

Despues; se va ala función de promedio de tipo float donde sabemos los valores de sus atributos de **a**  y **b,**

**prom**  va a llamar a la función llamada promedio junto con sus argumentos.

Nuestra función principal. **a** = 5, **b**=10, y **prom**  de tipo float pero no se le asigna nungun valor

Y **prom** retorna al inicio donde se manda a llamar la funcion. Se manda a llamar y se imprime un tipo de dato float por la letra **f** y enseguida  **prom**

**Tipos de variables**

**Variables Locales:**

Se declaran dentro de la función y sólo están disponibles durante su ejecución.

Se crean cuando se entra en ejecución una función y se destruyen cuando se termina.

**Variables globales**:

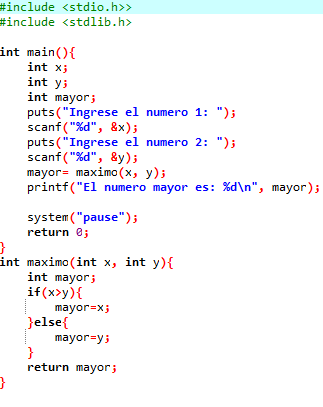
Se declaran fuera de las funciones. Pueden ser utilizadas por todas las funciones.

Existen durante toda la vida del programa**.**

**Ejercicios**

* Escribir una función que se lla.l9e maximo que reciba dos número por parámetros y que regrese el mayor de ellos.

Biblioteca

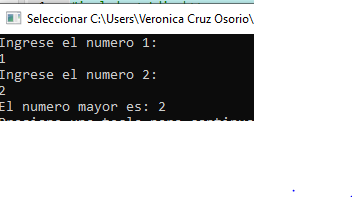


donde se manda a llamar el método máximo con atributos enteros x,y los mismo que se encuentran arriba.

Y se declara la variable mayor y donde se evalúa si el número es mayor se le asigna a **x** y si no es se le asigna a **y**

Se muestra en pantalla mensajes para ingresar el numero 1 y 2, donde muestra cual de los dos valores es el máximo.

Se declaran variables de tipo entero x, y, mayor



Se nuetsra cual es el máximo entre el numero 1 y 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:

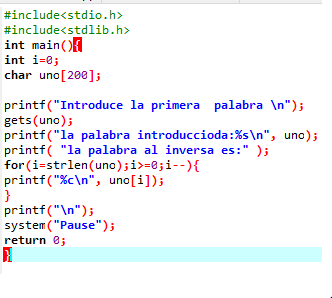
Entrada: Hola

Biblioteca

Se declara unavariable de tipo entero y una cadena de tipo char con nombre uno y con un tamaño de 200.

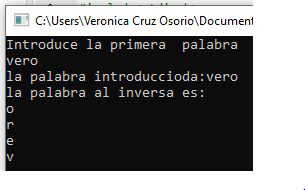
Se manda un mensaje en pantalla para que ingrese la palabra

Salida: aloH



Se manda a imprimir en pantalla

Donde se pone el valor de la cadena donde se muestre la palabra al revés, donde i tiene el valor de uno y que i vale 0 y existe un decremento por parte de la variable i

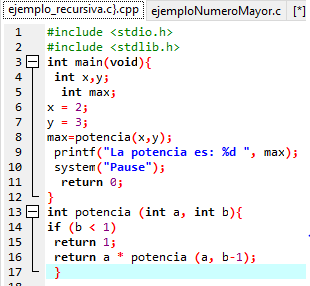


**Funciones recursivas**

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

Bibliotecas



Se manda a imprimir

se ejecuta un ciclo **if** donde< b es menor que 1 y hay

“b=3 es menor que 1”es verdadero.

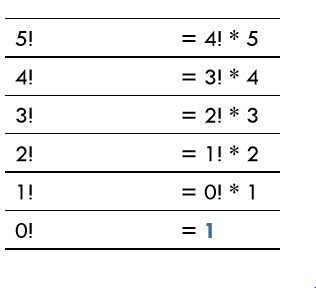
Primero se ejecuta lo que se encuentra en paréntesis que es a=2, **b=3-1=2)**

**y a=2 y b=2** ya resolviendo las operaciones. Resiviendo . y asi se va ejecuntando el ciclo if hasta llegar a 0. Al llegar al valor -1 b=0, despues va a ir regresando donde rotornados en a =2 y b=2, asi mismo con los demás valores

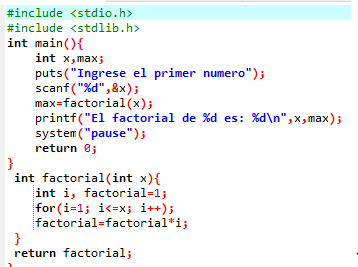
La variable max de tipo entero y que es igual a potencias y que tiene dos argumentos que son **x,y** donde se manda a llamar posteriormente donde potencias es de tipo entero y tiene dos argumentos de topo entero **a y b** , recordemos que a y b soy los valores anteriormente tenia x & y.

Las variables **x** & **y**  de tipo entero y que posteriormente se le asigna los valores x=2 & y=3

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



Bibliotecas



El el método factorial es donde se evalua el ciclo for y es donde se se va a multiplicar los valores el fila hacia a tras 1: ejemplo 5\*4, 4\*3, 3\*2, 2\*1

se muestra un mensaje donde se asigna el valor a **x,** y donde se manda a llamar a el método factorial y con sus atributos,

Se declara la variable x y max de tipo entero