

## Trabajo Práctico 2

Tipos de variables para cadenas de caracteres:

**char letra**// La variable letra, almacena un carácter.

**char cad[q]** // q, es la cantidad de caracteres máximos que puede albergar la variable cad en forma de arreglo de caracteres.

**string texto** //

La asignación en el momento de su definición permite agregar tanto en las variables de tipo **char[]** como en las variables de tipo **string**, un texto entre comillas doble (incluyendo espacios).

En las variables de tipo **char []** automáticamente agrega un carácter más que indica el fin de la cadena: **'\0'** (NULL)

Fuera del momento de la definición para ingresar una cadena de caracteres se utiliza la instrucción:

**cin.getline** (cad,q) // para variables de tipo **char[]**. Agrega al final el carácter **'\0'**

q es la cantidad de caracteres con que fue definida. Se puede utilizar **sizeof(cad)** para obtener **q** y con **strlen(cad)** se obtiene el tamaño de la cadena de caracteres ingresada.

**getline**(cin, texto) // para variables tipo **string**. No agrega el carácter **'\0'**. Con **texto.size()** o **texto.length()** averigua el tamaño de la cadena de caracteres ingresada.

Ambos tipos de variable pueden ser recorridos carácter por carácter utilizando un subíndice:

for (i=0; cad[i] != '\0'; i++) para la variable de tipo **char [ ]**

for (i=0; i<texto.size(); i++) para la variable de tipo **string** // es equivalente a **texto.length()**

Si tenemos

**char cadorigen[20]**={"Hola Mundo"}; // esta declaración agrega al final el carácter **'\0'** (NULL)

**char caddestino[20];**

**strcpy** (caddestino, cadorigen) copia en caddestino el contenido de cadorigen incluyendo el carácter **'\0'** (NULL)

Para dos variables de tipo **string**:

**String text1, text2**= "bienvenido ", la operación equivalente es:

text1=text2

**strcmp** (cad1, cad2) devuelve 0 si ambas cadenas son iguales, menor que 0 (-1) si la cadena 1 es menor que la cadena 2 y mayor que 0 (1) si la cadena 2 es mayor que la cadena 1.

Ej: strcmp("aaa", "bbb") devolverá -1 y strcmp("fff", "ccc") devolverá 1.

**strcat** (caddestino, cadorigen) añade el contenido de cadorigen a caddestino.

## Práctica

**Ej 1.** Escribir una función que recibe como parámetros una cadena de caracteres (char cad[]) y el tamaño máximo de la cadena int q, y que devuelve sin utilizar la función strlen(cad) la cantidad de caracteres de la cadena cad.

Verificar en el programa principal con la función strlen si el número que retornó la función es correcto.

**Ej2.** Definir dos cadenas de caracteres cad1 y cad2:

char cad1[41];

char cad2[41]="universidad nacional de avellaneda";

a) Copiar cad2 en cad1 tratando ambas cadenas como arreglos de caracteres.

Verificar después de la copia, mediante la instrucción strlen(), la longitud de ambas cadenas.

b) Hacer lo mismo pero ingresando el contenido de cad2 dentro del programa.

**Ej3.** Realizar una rutina que reciba 3 parámetros: un arreglo de n caracteres, n y un caracter, y muestre las posiciones donde se encuentra dicho carácter y cuantas veces se encuentra en el arreglo. Si no lo encuentra debe indicar que ese carácter no existe en el arreglo.

**Ej4.** Definir un arreglo de 200 caracteres. Ingresar un párrafo que no supere la cantidad máxima de caracteres del arreglo (el programa debe controlar este límite). El párrafo puede contener espacios y signos de puntuación (punto, coma, punto y coma, dos puntos, signos de admiración y de pregunta).

Se pide mostrar:

a) Cada una de las palabras que componen el párrafo y la cantidad de letras de cada una (**no** forman parte de las palabras los signos mencionados).

b) Cantidad de palabras que componen el párrafo

c) Longitud total del párrafo ingresado.

**Ej5.** Realizar una función que dado un arreglo de 45 elementos de tipo carácter que contiene una palabra, devuelva otro arreglo de 5 elementos enteros donde cada uno representa la cantidad de veces que se repite cada vocal en la palabra.

Para  
entregar