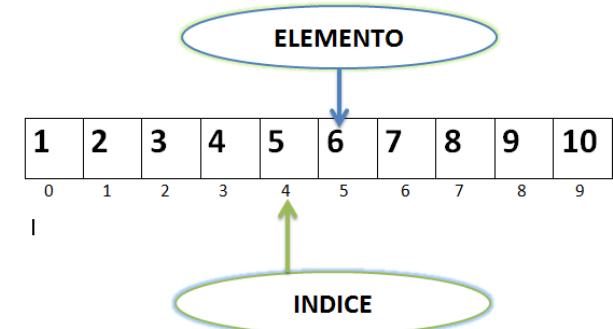


Asignatura:  
**(701) Programación II**

**CLASE 4**

**Cadena de caracteres: String**  
**Vectores asociados**  
**Ordenamiento Semi-Selección**

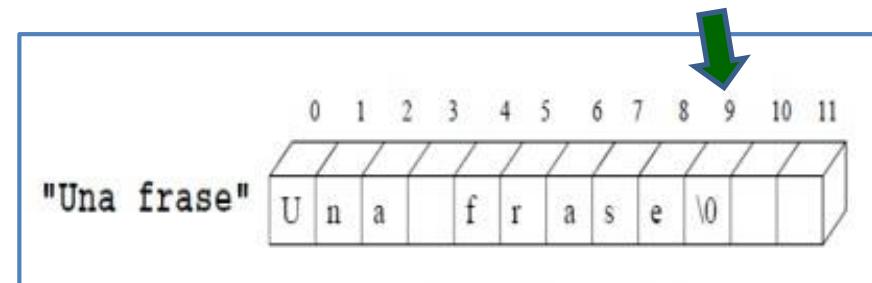
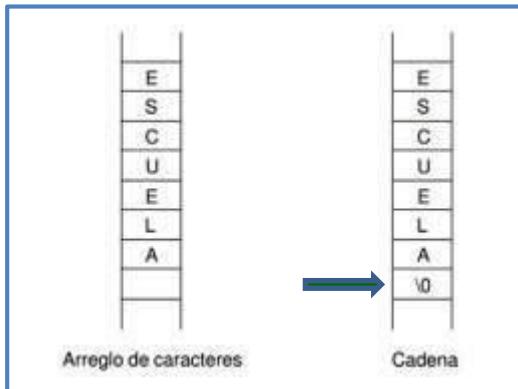


- Dejamos la **aclaración** que el compilador **DEV C++**, tiene algunas diferencias de tratamiento de caracteres, respecto a otros compiladores.

# Cadenas de caracteres

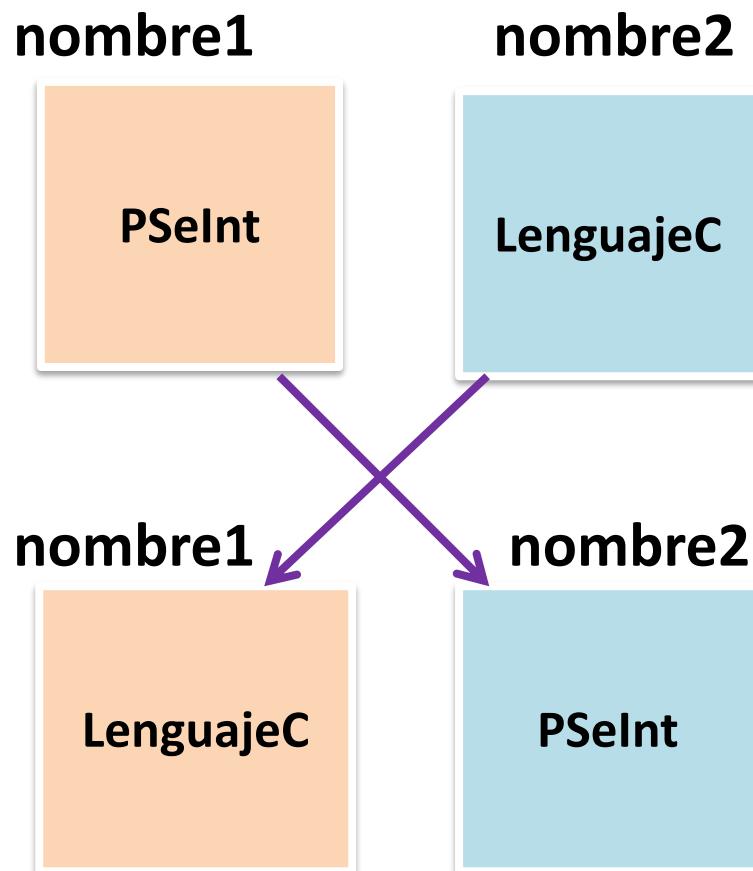
- Las cadenas de caracteres (*strings*)
- ✓ son un tipo particular de vectores,
- ✓ son variables que puede contener cero, uno o más elementos **char**,
- ✓ con la particularidad que tienen una **marca en el fin** del contenido (el carácter '**\0**').
- ✓ Lenguaje C, permite escribir las como texto dentro de **comillas dobles**.

```
char cadena1[]="Hola";           // cadena de 5 elementos  
  
char cadena2[]={‘H’,’o’,’l’,’a’,’\0’}; // Idem al anterior  
  
char cadenaVacia[]="";          // cadena de longitud 1
```



```
1 //|  
2 // manejo de CADENAS de caracteres.  
3  
4 #include <iostream>  
5 #include <conio.h>  
6 #include <stdio.h>  
7 #include <iomanip>  
8 #include <windows.h>  
9 #include <string.h> // para manejo de tira de caracteres  
10 using namespace std;  
11  
12 int main(void) {  
13     int largo, i, j ;  
14  
15     char cadena1[] = "Hola"; // cadena de 5 elementos ( 4 caracteres y marca de fin)  
16  
17     char cadena2[] = {'H','o','l','a','\0'}; // Idem al anterior  
18  
19     char cadenaVacia[] = ""; // cadena de longitud 1  
20  
21  
22     i=0;  
23     while (cadena1[i] !='\0' )  
24     {  
25         i++;  
26     }  
27     largo=i;  
28     cout << endl << "Cadena1: " << cadena1 << " ---> tiene " << largo << " caracteres" << endl << endl ;  
29  
30  
31     i=0;  
32     while (cadena2[i] !='\0' )  
33     {  
34         i++;  
35     }  
36     largo=i;  
37     cout << endl << "Cadena2: " << cadena2 << " ---> tiene " << largo << " caracteres" << endl << endl ;  
38  
39  
40     i=0;  
41     while (cadenaVacia[i] !='\0' )  
42     {  
43         i++;  
44     }  
45     largo=i;  
46     cout << endl << "CadenaVacia: " << cadenaVacia << " ---> tiene " << largo << " caracteres" << endl << endl ;  
47  
48  
49     getch();  
50     return (0);  
51  
52 }
```

## Intercambio entre 2 variables tipo de dato **string**



nombre1	nombre2	aux
PSeint	LenguajeC	PSeint
Lenguaje C	PSeint	

## Función “**strcpy**”

Sintaxis: **strcpy** (nombre cadena destino, nombre cadena origen)

- **int main(void) {**
- **char nombre1[30], nombre2[30], aux[30];**                   *// strings de 30 elementos.*
- **...**
- **// intercambio entre 2 variables string**
  
- **strcpy (aux,nombre1);**
- **strcpy (nombre1 , nombre2);**
- **strcpy( nombre2 , aux );**
- **...**
- **getch();**
- **return (0);**
- **}**

En lenguaje C, para asignar una cadena a un **array de caracteres (variable-destino)**, **no se puede utilizar el operador de asignación (=)**. (*No hacer: nombre1=nombre2;*)

Para ello, se puede utilizar la función **strcpy**.

La declaración de la **función strcpy** se encuentra en el archivo de cabecera **string.h**.

Por tanto, para poder usar dicha función en un programa, hay que agregar: **#include <string.h>**

## Flujo de entrada de datos: Funciones miembro **get** y **getline**

- Introduce un carácter en el flujo de datos de entrada,
- **incluso si es un espacio en blanco,**
- Lee, hasta un carácter menos, que el número máximo de caracteres especificado en la definición de la variable, o hasta cuando lea un delimitador.
- Se inserta un carácter nulo para terminar la cadena de caracteres.

Se debe agregar la librería: **#include <string.h>**

### Sintaxis:

**cin.getline (variable-destino-de-lo-que-lea, longitud-destino, '\n');**

- 1) El nombre de la variable que va a contener el string.
- 2) Longitud. La cantidad de caracteres que se deben ingresar (*nunca mayor que la longitud del string*) o un valor estimativo igual que la longitud definida para el string.
- 3) El carácter que el usuario va usar como final de la cadena. Por lo general es el 'enter' que se representa como '\n' (barra invertida n).

Ejemplo: **cin.getline (palabra,50,'\\n');**

## Ejercicio:

Detecta cantidad de caracteres del texto leído y lo imprime al revés.

```
22
23 int main(void) {
24
25     int largo, i, j, mitad, izq, der, vez, iguales ;
26
27     char texto[50]; // defino una cadena de caracteres
28
29     cout << " Ingrese un TEXTO, puede ser palabra o frase con blancos intermedios: " << endl;
30
31     cin.getline (texto,50,'\n'); // lee palabras y blancos hasta el fin de linea      // si se lee con cin(texto) --> solo ingresa hasta el primer caracter blanco.
32
33
34 // Parte 1: detecta cantidad de caracteres de la palabra ingresada. La informa al revés.
35
36     i=0;
37     while (texto[i] !='\0' )
38     {
39         i++;
40     }
41     largo=i;
42     cout << endl << "El texto ingresado tiene " << largo << " caracteres" << endl << endl ;
43
44     cout << "El texto al revés es: ";
45
46     for(j=largo-1; j >=0; j--)
47     {
48         cout << texto[j];
49
50     }
51     cout << endl;
52
```

```
cin.getline (texto,50,'\\n');
```

// lee palabras y blancos hasta el fin de línea

// si se lee con `cin(texto);` --> *solo ingresa hasta el primer carácter blanco.*

## **Ejercicio**

- ✓ para práctica de vectores asociados,
- ✓ para analizar un algoritmo de ordenamiento y manejo de strings:

- **Semi-Selección**, observar que a medida que va buscando el menor, lo va llevando a la primer posición del rango de búsqueda.

*OJO:* Hace MUCHOS más movimientos que el ordenamiento por Selección.

- Completar el programa, agregándole que imprima el vector después de cada modificación.
- Agregarle Punto extra: Informar los valores de las ventas de más o igual a un VALOR, (*dato que se solicitará al usuario*), y el/los correspondiente/s nombres de los vendedores que realizaron esas ventas.

```

10
11 int main(void) {
12     float venta[100], auxV, suma=0;
13     char vendedor[100][30], aux[30];           // vector de strings de 100 elementos. Cada string tiene 30 posiciones
14     int i, j, n, w, tope ;
15     cout<<"ingrese la cantidad de Vendedores: " << endl;
16     cin>>n;
17     for(i=0; i<n; i++)
18     {
19         cout<<"Vendedor "<< i+1 << endl;
20         cout<<"Nombre Vendedor:";
21         cin>> vendedor[i];
22         cout<<"Importe de su Venta:";
23         cin>> venta[i];
24         cout << " ";
25     }
26
27     for(i=0; i<n; i++)
28     {
29         suma += venta[i];           // Calcula la SUMA de los importes de ventas para calcular el Promedio
30     }
31     for (w=0; w< n; w++) {           // imprime el vector ORIGINAL
32         cout<< " "<< venta[w];
33     }
34     cout<< endl;
35
36     for ( i=0; i<n-1; i++) {
37
38         for ( j=i+1; j<n; j++) {
39
40             if (venta[i]>venta[j])
41             {
42                 auxV=venta[i];           // ordena el vector de Importes de La Venta ( de menor a mayor)
43                 venta[i]=venta[j];
44                 venta[j]=auxV;
45
46                 strcpy (aux,vendedor[i]); // realiza el MISMO cambio en el vector ASOCIADO de Nombres de Vendedores
47                 strcpy(vendedor[i],vendedor[j]);
48                 strcpy(vendedor[j],aux);
49
50             for (w=0; w< n; w++) {       // imprime el vector para ver la modificación en el orden
51                 cout<< " "<< venta[w];
52             }
53             cout<< endl;
54
55         }
56     }

```