

## Cadenas

**Cadena:** es un array de caracteres (char) terminado en un carácter nulo (\0)

Array de caracteres

A	B	C		
---	---	---	--	--

Cadena de Caracteres

A	B	C	\0	
---	---	---	----	--

Las cadenas se declaran como los arrays. Se inician con comillas.

```
char texto[5] = "ABC"; //contiene cadenas de 4 caracteres como máximo
```

### 1 Lectura y escritura de cadenas desde consola

leer	cin >> nombre del array que recibe la cadena	cin >> texto;
Escribir	cout << nombre del array que contiene la cadena	cout << texto;

**Problemas:**

**cin no hace comprobación de límites**, de manera que es posible que el usuario introduzca más caracteres que los que pueda almacenar la cadena.

**Si intento introducir desde teclado la cadena "Esto es un prueba"** C++ deja de leer encuentra el primer espacio en blanco y por lo tanto sólo habrá almacenado "Esto".

Para solucionarlo se utiliza la función de biblioteca gets() que lee una cadena del teclado hasta que se pulsa intro, el formato general de gets es: gets(nombre\_array);

```
ej: #include <iostream.h>
#include <stdio.h>
```

```
void main ()
{   char texto[75];
    gets(texto);
    cout << texto;
}
```

Observación C++ también tiene funciones orientadas a objetos que le permiten leer cadenas desde teclado.

**Las cadenas no se asignan como el resto de las variables.**

```
Texto = "Hola"; // error
```

## 2 Algunas funciones de cadenas (string.h)

Función	Cabecera de la función o prototipo	descripción
Strlen	size_t strlen(const char *);	Devuelve la longitud de la cadena s
Strcpy	char * strcpy (char *dest, const char *fuente);	Copia la cadena "fuente" en la cadena "destino"
Strcmp	int strcmp(const char *s1, const char *s2);	Compara las cadenas s1 y s2 y devuelve: 0 si s1 = s2 <0 si s1 < s2 >0 si s1 > s2
Strcat	char *strcat(char *dest, const char *src);	concatena la cadena "fuente" al final del "destino"

Función	Código de ejemplo	Visualización por consola
strlen	<pre>#include &lt;iostream.h&gt; #include &lt;string.h&gt; void main () {     char texto[75] = "Hola y adios";     int longitud = 0;     longitud = strlen (texto);     cout &lt;&lt; longitud; }</pre>	12
strcpy	<pre>#include &lt;iostream.h&gt; #include &lt;string.h&gt; void main () {     char texto_origen[75] = "Hola y adios";     char texto_destino[25] ;     strcpy (texto_destino, texto_origen);     cout &lt;&lt; texto_destino; }</pre>	Hola y adios
strcmp	<pre>#include &lt;iostream.h&gt; #include &lt;string.h&gt; void main () {     char texto_1[75] = "Hola y adios";     char texto_2[25] = "hola y adios";     int resultado;     resultado = strcmp (texto_1, texto_2);     cout &lt;&lt; resultado; }</pre>	-1
	<pre>#include &lt;iostream.h&gt; #include &lt;string.h&gt; void main () {     char texto_1[75] = "Hola y adios";     char texto_2[25] = "hola y adios";     int resultado;     resultado = strcmp (texto_2, texto_1);     cout &lt;&lt; resultado; }</pre>	1

	<pre> #include &lt;iostream.h&gt; #include &lt;string.h&gt; void main () {     char texto_1[75] = "Hola y adios";     char texto_2[25] = "Hola y adios";     int resultado;     resultado = strcmp (texto_2, texto_1);     cout &lt;&lt; resultado; } </pre>	0
strcat	<pre> #include &lt;iostream.h&gt; #include &lt;string.h&gt; void main () {     char texto_1[75] = "Hola ";     char texto_2[25] = "y adios";     strcat (texto_1, texto_2);     cout &lt;&lt; texto_1; } </pre>	Hola y adios