



Programación Avanzada



Pontificia Universidad
JAVERIANA
Colombia

Cadenas de caracteres

En lenguaje C++ existen tres formas básicas de crear variables para manejo de cadenas de caracteres

- **char variable[n]**
- **string variable**
- **char *variable**

Independiente de la forma de declaración para acceder o cambiar el valor de una posición de la cadena la manejamos como un arreglo.

Ej:

n=2;

variable[n]='c';

cout << variable[n];

Formas de declarar variables de cadena de caracteres:

Ej:

- `char cad1[20]={'h','o','l','a',' ','m','u','n','d','o','\0'};`
- `char cad2[]={ 'h','o','l','a',' ','m','u','n','d','o','\0'};`
- `char cad3[]="hola mundo";`
- `char cad4[20]= "hola mundo";`
- `char *ptrCad1="hola mundo";`
- `string strCad2="esta es una cadena muy larga declarada como variable string y no como arreglo de char";`
- `string strCad1="hola mundo";`



Pontificia Universidad
JAVERIANA
Colombia

Cadenas de caracteres – funciones para arreglos de caracteres

<u>Función</u>	<u>Sintaxis</u>	<u>Significado</u>
strcat	<code>char *strcat(char *destino, const char *origen);</code> Ej: <code>strcat(cad1, " ");</code> <code>strcat(cad1, cad2);</code>	Agrega una copia de la cadena origen al final de la cadena destino.
strchr	<code>char *strchr(char *cad, char c);</code> Ej: <code>strchr(cad1, 'a');</code>	Busca la primera ocurrencia del caracter contenido en la variable "c" y devuelve una nueva cadena a partir de esa posición. Si no encuentra el carácter buscado, genera error en tiempo de ejecución.
strcmp	<code>int strcmp(const char *cad1, const char *cad2);</code> Ej: <code>strcmp(cad1, cad3)</code>	Compara uno a uno los caracteres de ambas cadenas hasta completarlos todos o hasta encontrar una diferencia. Devuelve cero (0) si ambas cadenas son iguales, un valor negativo si la primera cadena es menor que la segunda y un valor positivo si la primera es mayor que la segunda.
strcpy	<code>char *strcpy(char *destino, const char *origen);</code> Ej: <code>strcpy(cad5, cad4);</code>	Copia todos los caracteres de la cadena origen en la cadena destino, incluso el caracter de terminación. La cadena destino debe ser lo suficientemente grande para almacenar la información de la cadena origen, de lo contrario genera error en tiempo de ejecución
strlen	<code>int strlen(const char *cad);</code> Ej: <code>strlen (cad5)</code>	Devuelve el número de caracteres de la cadena "cad". No cuenta el caracteres de terminación.

Cadenas de caracteres – funciones para arreglos de caracteres

<u>Función</u>	<u>Sintaxis</u>	<u>Significado</u>
strlwr	<code>char *strlwr(char *cad);</code> Ej: <code>strlwr(cad7);</code>	Convierte las letras contenidas en la cadena "cad" en minúsculas.
strrchr	<code>char *strrchr(char *cad, char c);</code> Ej: <code>char *cad9=strrchr(cad7, 'a');</code>	Busca la última ocurrencia del caracter contenido en la variable "c" y devuelve una nueva cadena a partir de esa posición.
strrev	<code>char *strrev(char *cad);</code> Ej: <code>strrev(cad7);</code>	Invierte los caracteres contenidos en la cadena "cad".
strstr	<code>char *strstr(const char *cad1, const char *cad2);</code> <code>strstr(cad10,cad11);</code>	Busca la primera ocurrencia de la cadena "cad2" en la cadena "cad1", y devuelve una nueva cadena a partir de esa posición.
strupr	<code>char *strupr(char *cad);</code> Ej: <code>strupr(cad8);</code>	Convierte las letras contenidas en la cadena "cad" en mayúsculas.



Ejemplo 1.cpp

Cadenas de caracteres – funciones y metodos string

Para entender los métodos de las variables string debemos conocer el concepto de **iterador**. Un **iterador** es una especie de puntero que es utilizado por un **algoritmo** para recorrer los elementos almacenados en un **contenedor**; para nuestro caso, la variable tipo string.

begin: retorna un iterador desde el principio de una variable string

end: retorna un iterador desde el final de una variable string

rbegin: retorna un iterador tomando como inicio el fin de un variable string

rend: retorna un iterador tomando como fin el principio de un variable string

size: Devuelve la longitud de la cadena, en términos de bytes.

length: Devuelve la longitud de la cadena, en términos de bytes.

max_size: Devuelve la máxima longitud de la cadena, en términos de bytes.

capacity: Devuelve el tamaño del espacio de almacenamiento asignado actualmente a la cadena, expresado en bytes

clear: Borra el contenido de la cadena, que se convierte en una cadena vacía (con una longitud de 0 caracteres).

empty: Valida si la cadena de caracteres esta vacía



Ejemplo 2.cpp



Ejemplo 3.cpp



Pontificia Universidad
JAVERIANA
Colombia

Cadenas de caracteres – funciones y metodos string

resize: Cambia el tamaño de la cadena a una longitud de n caracteres. Los caracteres a partir de n+1 se pierden.

shrink_to_fit: Solicita al string que reduzca su capacidad para ajustarse a su tamaño.

at(pos): Devuelve una referencia al carácter en la posición pos de la cadena. Funciona similar a **cadena[pos]**.

back: devuelve una referencia al ultimo carácter de la cadena.

front: devuelve una referencia al primer carácter de la cadena.

append: Extiende la cadena agregando caracteres adicionales al final de su valor actual:

assign: Asigna un nuevo valor a la cadena, reemplazando su contenido actual.

insert: Inserta caracteres adicionales en la cadena justo antes del carácter indicado por pos (op):

erase: Borra parte de la cadena, reduciendo su longitud

replace: Reemplaza parte de un string

find: Busca en la cadena la primera aparición de la secuencia especificada por los argumentos. Devuelve -1 si no encontró la secuencia indicada

substr: devuelve un objeto de cadena construido teniendo como valor inicializado una copia de una subcadena.

compare: Compara el valor del objeto de cadena (o una subcadena) con la secuencia de caracteres especificada por sus argumentos. Si es cero la comparación es correcta



Ejemplo 4.cpp



Ejemplo 5.cpp



Ejemplo 6.cpp



Ejemplo 7.cpp



Ejemplo 8.cpp



Ejemplo 9.cpp



Ejemplo 10.cpp



Ejemplo 11.cpp



Ejemplo 12.cpp



Ejemplo 13.cpp



Ejemplo 14.cpp



Pontificia Universidad
JAVERIANA
Colombia