



UNIDAD CURRICULAR: ALGORITMICA Y PROGRAMACIÓN

UNIDAD VIII. TRATAMIENTO DE CADENAS DE CARACTERES

CONTENIDO:

[Las cadenas](#)

[Funciones asociadas y operaciones con cadenas](#)

[Ejercicios Resueltos](#)

[Referencias Bibliográficas](#)





UNIDAD VIII

TRATAMIENTO DE CADENAS DE CARACTERES

Una cadena es un tipo de dato compuesto por un conjunto de caracteres. Estos caracteres incluyen letras, números y caracteres especiales. También son llamadas strings, tira de caracteres o cadenas de caracteres.

En C++ las cadenas son tratadas como un arreglo de caracteres o char.

FUNCIONES ASOCIADAS Y OPERACIONES CON CADENAS:

Las operaciones con cadenas más usuales son:

- Asignación de cadenas
- Cálculo de la longitud de una cadena
- Comparación de cadenas
- Unión de cadenas.
- Cambio a mayúsculas
- Cambio a minúsculas

Existen otras operaciones, tales como, substraer subcadenas, reemplazar subcadenas, contar el número de ocurrencias de una subcadena en una cadena, determinar si contiene sólo letras o sólo números, entre otras.

A continuación se presentan las funciones de cadenas más utilizadas en C++, las cuales están implementadas en el archivo de biblioteca estándar **string.h**.

Asignación de Cadenas: Se utiliza para copiar una cadena de caracteres (fuente) en el lugar de otra (destino).

Sintaxis:

strcpy(destino, fuente)

Ejemplo:

`strcpy(nombre, "Juan Torres");`

En este caso la variable *nombre* tomará el valor *Juan Torres*.



Cálculo de la longitud de una cadena: Dada una cadena se puede determinar la cantidad de caracteres que posee.

Sintaxis:

strlen(cadena)

Ejemplo:

```
longitud=strlen(nombre);
```

Siguiendo con el ejemplo anterior, la variable longitud tendrá el valor de 11 que corresponde al número de caracteres que tiene la cadena *Juan Torres*.

Comparación de cadenas: Al comparar dos cadenas es posible obtener tres resultados: Un valor menor a cero que indica que la primera cadena es menor que la segunda cadena. Un valor mayor a 0 que indica que la primera cadena es mayor que la segunda. Si el resultado es 0 indica que las dos cadenas son iguales.

Sintaxis:

strcmp(primer_cadena,segunda_cadena)

Ejemplo:

```
strcpy(cadena1, "ABC");
```

```
strcpy(cadena2, "DEF");
```

```
resultado=strcmp(cadena1,cadena2);
```

El valor de la variable *resultado* será un número menor a 0, ya que *cadena1* es menor que *cadena2*.

Unión de cadenas: Se pueden concatenar dos cadenas de caracteres. Para ello, la primera cadena se une con la segunda.

Sintaxis:

strcat(primer_cadena,segunda_cadena)

Ejemplo:

```
strcpy(cadena1, "ABC");
```

```
strcpy(cadena2, "DEF");
```

```
strcat(cadena1,cadena2);
```

En este caso, la variable *cadena1* contendrá finalmente la cadena *ABCDEF*.



Cambio a mayúsculas: En una cadena se pueden convertir las letras que contiene en mayúsculas.

Sintaxis:

strupr(cadena)

Ejemplo:

```
strcpy(nombre, "Juan Torres");  
strupr(nombre);
```

El valor de la variable *nombre* será *JUAN TORRES*.

Cambio a minúsculas: En una cadena se pueden convertir las letras que contiene en minúsculas.

Sintaxis:

strlwr(cadena);

Ejemplo:

```
strcpy(nombre, "Juan Torres");  
strlwr(nombre);
```

El valor de la variable *nombre* finalmente será *juan torres*.

EJERCICIOS RESUELTOS

a) Realice un programa que lea una cadena por teclado e indique su longitud.

```
#include <iostream>  
#include <string.h>  
using namespace std;  
char cadena[20];  
int longitud;  
int main()  
{ cout << "Ingrese la cadena de texto: ";  
  cin>> cadena;  
  longitud=strlen(cadena);  
  cout <<"La cadena tiene "<<longitud<< " caracteres";  
  return 0;  
}
```

b) Realice un programa que lea en una variable el nombre de una persona y en otra variable el apellido. Luego en la variable nombre agregue el apellido y muéstrelo.

```
#include <iostream>  
#include <string.h>
```



```
using namespace std;
char nombre[40], apellido[20] ;
int main()
{ cout << "Ingrese el nombre: ";
  cin>> nombre;
  cout << "Ingrese el apellido: ";
  cin>> apellido;
  strcat(nombre,apellido);
  cout <<"El nombre completo es: "<<nombre;
  return 0;
}
```

c) Realice un programa que lea dos cadenas e indique si son iguales, o cuál de las dos es la menor.

```
#include <iostream>
#include <string.h>
using namespace std;
char cadena1[20], cadena2[20] ;
int resultado;
int main()
{ cout << "Ingrese la primera cadena: ";
  cin>> cadena1;
  cout << "Ingrese la segunda cadena: ";
  cin>> cadena2;
  resultado=strcmp(cadena1,cadena2);
  if (resultado < 0)
    cout <<"La cadena "<<cadena1 << " es menor";
  else
    if (resultado>0)
      cout <<"La cadena "<<cadena2 << " es menor";
    else
      cout <<"Las cadenas son iguales ";
  return 0;
}
```

d) Realice un programa que lea dos palabras distintas y muestre la palabra menor en minúscula y la mayor en mayúscula.

```
#include <iostream>
#include <string.h>
using namespace std;
char cadena1[20], cadena2[20] ;
int resultado;
int main()
{ cout << "Ingrese la primera palabra: ";
  cin>> cadena1;
  cout << "Ingrese la segunda palabra: ";
  cin>> cadena2;
```



```
resultado=strcmp(cadena1,cadena2);
if (resultado < 0)
    cout << strupr(cadena2)<<strlwr(cadena1);
else
    if (resultado>0)
        cout << strupr(cadena1)<<strlwr(cadena2);
    else
        cout << "Las palabras son iguales ";
return 0;
}
```

e) Realice un programa que lea dos apellidos distintos y los muestre en mayúscula, en orden alfabético y ascendente.

```
#include <iostream>
#include <string.h>
using namespace std;
char apellido1[20], apellido2[20] ;
int resultado;
int main()
{ cout << "Ingrese el primer apellido: ";
  cin>> apellido1;
  cout << "Ingrese el segundo apellido: ";
  cin>> apellido2;
  resultado=strcmp(strupr (apellido1), strupr (apellido2));
  if (resultado < 0)
      cout << strupr(apellido1)<<strupr(apellido2);
  else
      if (resultado>0)
          cout << strupr(apellido2)<<strupr(apellido1);
      else
          cout << "Los apellidos son iguales ";

  return 0;
}
```

REFERENCIAS BIBLIOGRÁFICAS

Bassard, G y Bratley, P. (2010). Fundamentos de algoritmia. Prentice-Hall.

Joyanes, L. (2008). Fundamentos de programación. Algoritmos , Estructuras de datos y objetos. Mc Graw Hill. Tercera edición.

Joyanes, L. y Zahonero, I. (2005). Programación en C. Metodología, algoritmos y Estructura de datos. Mc Graw Hill. Segunda Edición

Martí, N. y Ortega, Y. (2004). Estructuras de datos y Métodos Algorítmicos. Ejercicios Resueltos. Pearson Education.



Programa Nacional de Formación de Informática

Material Educativo Computarizado de Algorítmica y Programación