Algorítmica I

Sesión 07

Manejo de Cadenas de caracteres

Sesion 07

Manejo de Caracteres

Cadena de Caracteres

Library String.h

- Tipo de datos CHAR
- Tipo de datos STRING
- Funciones de cadenas
  - Copiar cadenas
  - Añadir cadenas a otras
  - Buscar carácter dentro de una cadena
  - Longitud de cadena

Cadena de Caracteres

- Arreglo en el que en cada elemento es de tipo char.
- Los caracteres de una cadena se colocarán uno a continuación del otro en los elementos del arreglo, a partir del primero.
- Para poder saber donde terminan los caracteres válidos en el arreglo, se debe colocar un valor de terminación al final de la secuencia de caracteres este valor es cero.
- Este valor se le puede proporcionar al arreglo simplemente '\0'.

DECLARACIÓN

Una cadena de caracteres es un arreglo de tipo char, por lo tanto una cadena de caracteres se declara de la siguiente manera:

```
char identificador [límite];
```

Por ejemplo:

```
char nombre[20];
```

Donde "nombre" es una cadena de caracteres que puede albergar hasta 19 caracteres, no olvidar que se requiere un espacio para el delimitador de la cadena (cero o '\0'). por eso no se pueden albergar en ella 20 caracteres.

INICIALIZACIÓN:

Como cualquier arreglo, una cadena de caracteres se puede ser inicilizado en el momento de su declaración. Esta opracón se realiza de la siguiente manera:

- char nomb1[10] = { 72, 111, 108, 97,   0, 43,   9, 123,   10, 45}; ó
- char nomb2[10] = {'H', 'o',   'l', 'a', '\0', '+', '\t',   '{', '\n', 'A'};

Inicialización

char nomb3[10] = {'H', 'o', 'l', 'a'};

Esto signfica que: nomb3[0] = 'H', nomb3[1] = 'o', nomb3[2] = 'l', nomb3[3] = 'a', nomb3[4] = 0 ó '\0', nomb3[5] = 0 ó '\0', etc., lo que significa que la cadena está perfectamente delimitada.

Otra opción que nos da el compilador para inicializar una cadena de caracteres es haciendo uso de una secuencia de caracteres encerrados entre comillas ("..."), como se muestra a continuación:

char nomb4[10] = "Hola";

Esto es equivalente a la inicialización de la variable nomb3. Incluso se le ha colocado el delimitador en la quinta posición.

LECTURA Y ESCRITURA DE CADENAS DE CARACTERES

funciones scanf y printf. (Lenguaje C)

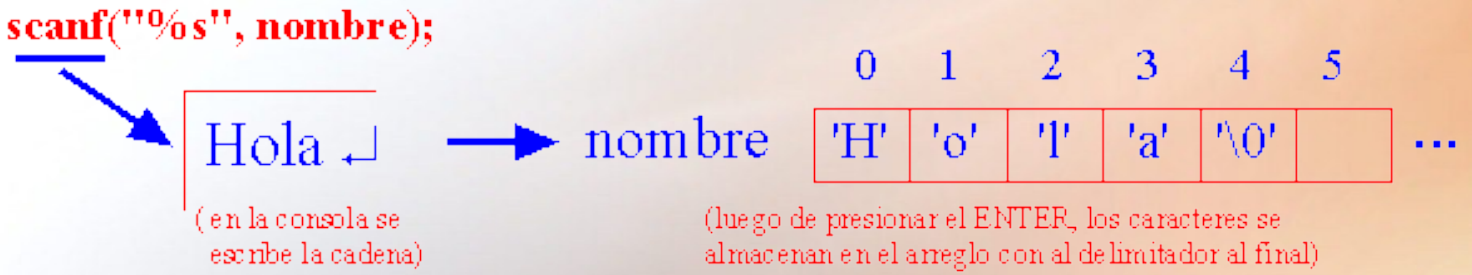
La función **scanf**, permite el ingreso de datos desde la consola. Esta misma función, empleando el especificador de formato **%s**, permite el ingreso de cadenas de caracteres.

ejemplo:

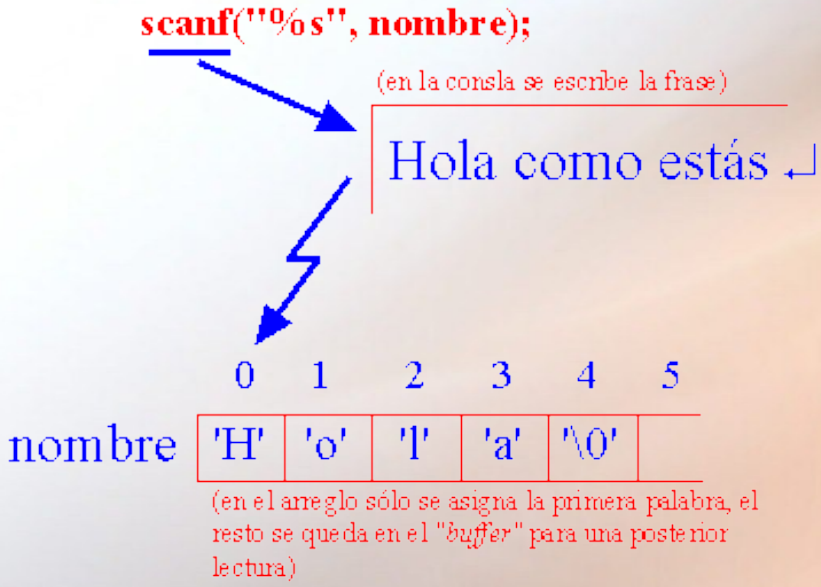
```
char nombre[20];
scanf("%s", nombre);
```

```
int valor;
scanf("%s", &valor);
```

La función scanf irá tomando uno a uno los caracteres del buffer de entrada, y los irá colocando consecutivamente en el arreglo a partir del primer elemento. Al fnal colocará el delimitador 0 ó '\0' en el siguiente elemento, imediatamente después del último caracter ingresado.

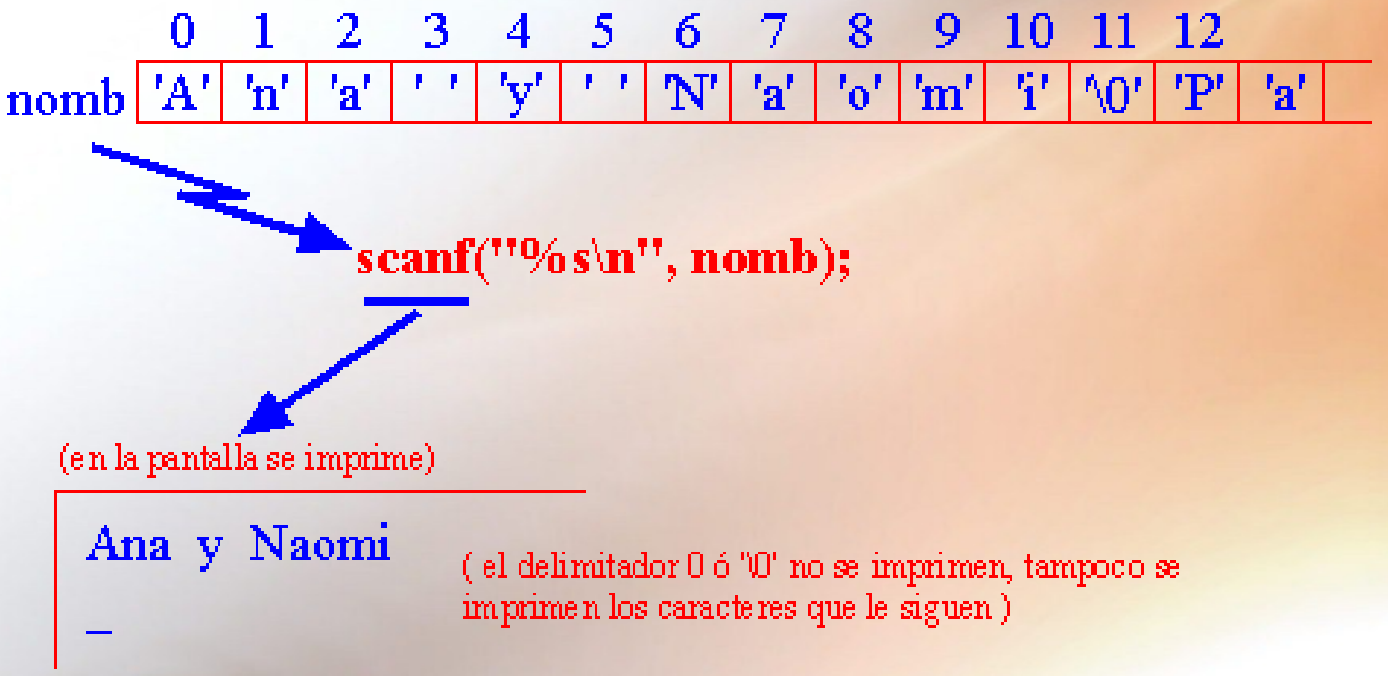


- La exploración de un dato se realiza hasta encontrar un "espacio" (un caracter blanco, un caracter de tabulación o un caracter de cambio de línea). Es por esta razón que empleando el especificador de acceso %s en la función scanf sólo se podrán leer palabras sueltas y no frases completas.



printf

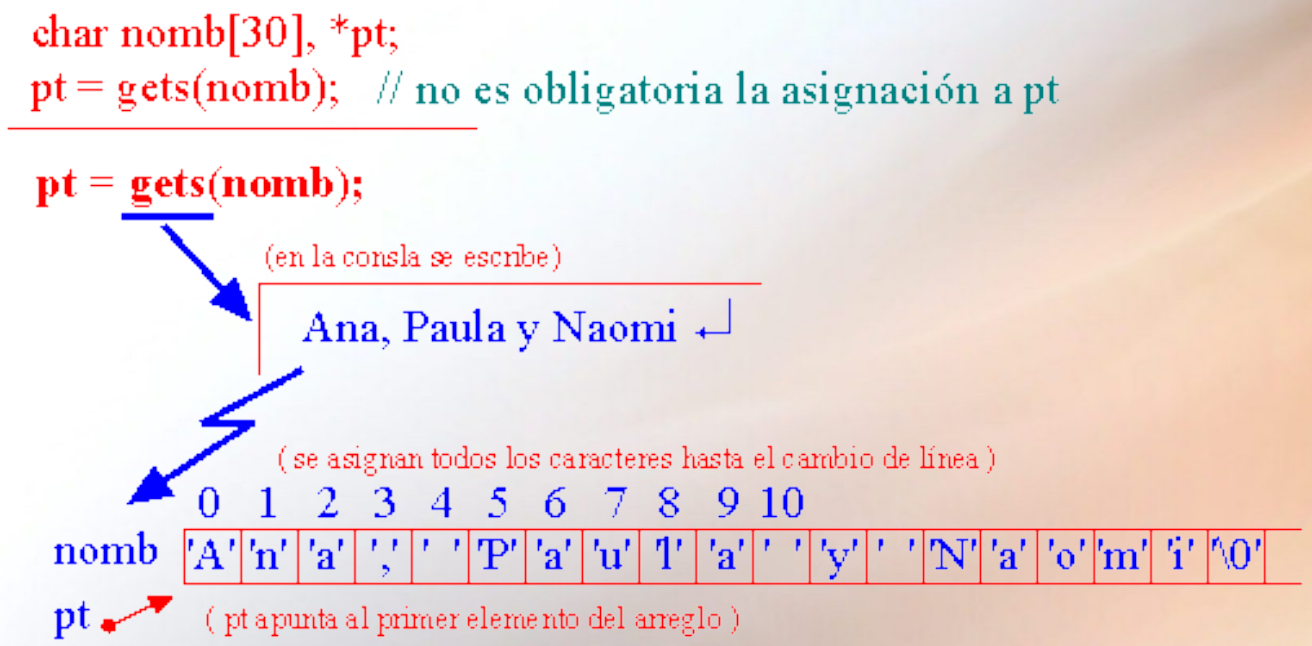
La función printf permite, empleando el especificador de formato %s, la impresión del contenido de un arreglo de tipo char, como caracteres en la consola. La función irá tomando cada uno de los elementos del arreglo e irá mostrando en la pantalla cada uno de los caracteres correspondientes, hasta encontrar el delimitador de la cadena (0 ó '\0'). Si el delimitador no se encontrara presente en el arreglo, se seguirían imprimiendo caracteres contenidos en los bytes contiguos hasta encontrar uno.



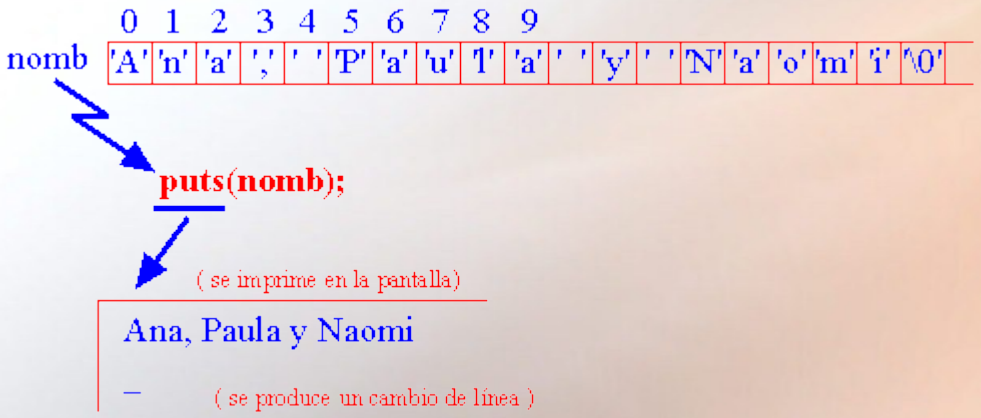
Función gets y puts

La función gets recibe como parámetro un arreglo de tipo char, en él se almacenará los caracteres provenientes de la consola.

A difierencia de prints la lectura con gets no se detendrá cuando se encuentre un espacio en blanco sino hasta encontrar un caracter de cambio de línea.

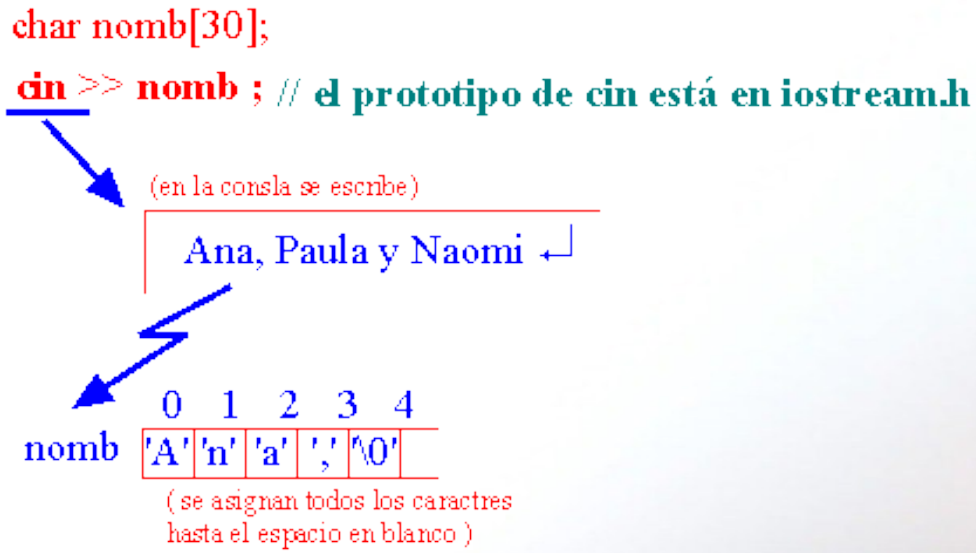


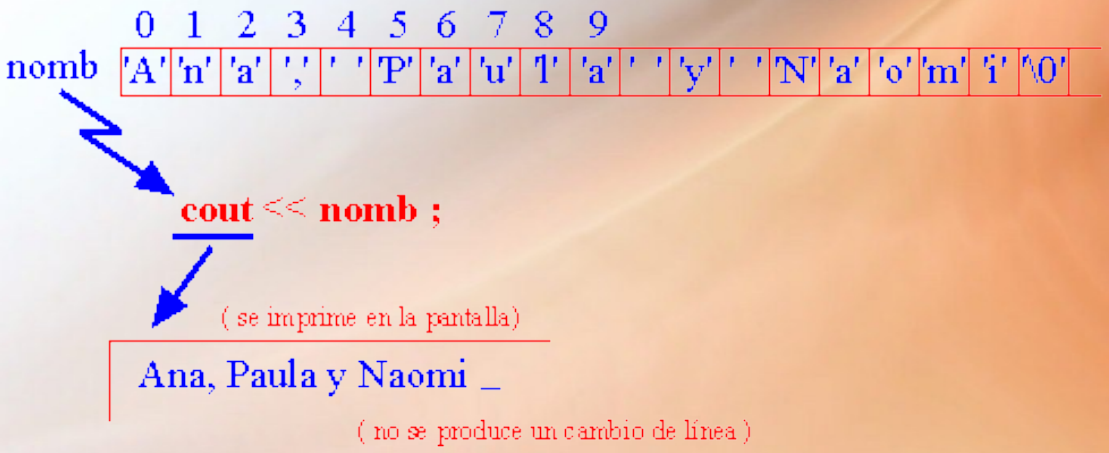
- La función puts es equivalente a scanf con la diferencia que cuando puts termina de imprimir los caracteres, envia a la pantalla un caracter de cambio de línea.
- La funció́m puts devuelve un valor entero positivo si la operación tiene éxito y un valor negativo (por lo general EOF) si hubo un error de lectura.



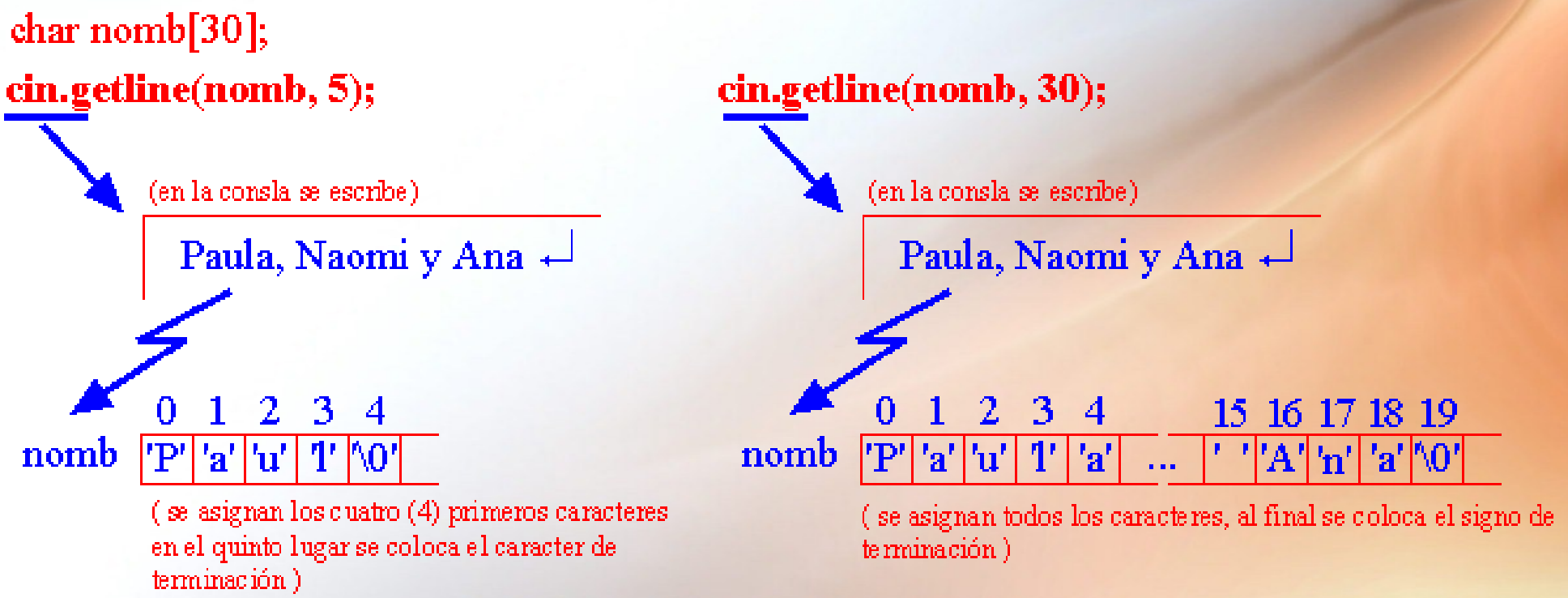
cin / cout

Finalmente se pueden emplear los objetos cin y cout.
El uso de ambos objetos tiene un efecto igual al del %s en las funcions scanf y printf.





- También existe un método asociado a cin que permite realizar una operación similar a **gets**, este método se denomina **getline**



Limpieza de Buffer

```
#include <iostream>

using namespace std;

int main (){
    //Declaracion de variables
    string Nombre;
    int Edad;

    cout<<endl<<"Ingrese su Edad: "<<endl; //Pide al usuario datos
    cin>>Edad;                             //El usuario ingresa los datos
    cin.ignore();                         //<-------Limpiamos el Buffer de tal manera que el programa pueda continuar
    //fflush(stdin);
    cout<<"Ingrese su Nombre: "<<endl;     //Pide al usuario datos
    getline(cin,Nombre);                   //El usuario ingresa los datos

    //El programa imprime los datos
    cout<<endl<<"Su Nombre es: "<<Nombre<<endl<<"Y su Edad es: "<<Edad<<endl;

    return 0;
}
```

Operaciones con Cadenas (string.h)

Longitud (strlen): Obtiene la longitud de una cadena

Sintaxis

strlen(cadena) ;

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

void main()
{
    char nombre[30];
    int tamano;
  cout<<"¿Cuál es tu nombre?\n";
  cin>>nombre;
  tamano = strlen( nombre );
    cout<<"Tu nombre tiene
"<<tamano<<"letras";
```

Comparación (strcmp)

Permite comparar dos cadenas e indica si son exactamente iguales

Sintaxis

strcmp (cadena1, cadena2);

Esta función devuelve un valor de acuerdo al resultado de la comparación.

Devuelve:

0	si la dos cadenas son exactamente iguales
Mayor a 0	si la cadena1 es mayor a la cadena2
Menor a 0	si la cadena1 es menor que la cadena2

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

void main()
{
   char contrasena[30], reContrasena[30];
   int resultado;
   cout<<"Escribe tu contraseña\n";
   cin>>contrasena;
   cout<<"Re escribe tu contraseña\n";
   cin>>reContrasena;
   resultado = strcmp(contrasena, reContrasena);
   if ( resultado == 0 )
      cout<<"La contraseña es aceptada";
   else

      cout<<"La contraseña no coincide";

}
```

Copia (strcpy)

Permite copiar todo el contenido de una cadena a otra, reemplazando el contenido de la cadena destino

Sintaxis

strcpy(cadenaDestino, cadenaOrigen);

Todo el contenido de la cadenaOrigen se copia a la cadenaDestino, si esta última tuviera algún valor este se reemplaza.

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

void main()
{
    char origen[30], copia[30];
    cout<<"¿Qué día es hoy? \n";
    cin>>origen;
    strcpy(copia, origen);
    cout<<"Hoy es "<<copia;
}
```

Concatenación (strcat)

Permite juntar o concatenar dos cadenas una a continuación de la otra.

Sintaxis

strcat(cadenaDestino, cadenaOrigen);

Todo el contenido de la cadenaOrigen se añade al final de la cadenaDestino

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

void main()
{
char nombre[30], apellido[30];
cout<<"¿Cuál es tu nombre? \n";
cin>>nombre;
cout<<"¿Cuál es tu apellido paterno\n";
cin>>apellido;
strcat(nombre, " "); //Se le añade un espacio en blanco
strcat(nombre, apellido);
cout<<"Tu nombre completo es "<<nombre;
}
```

strupr

- Convierte las letras contenidas en una cadena a mayúsculas

strupr(cadena);

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

void main()
{
   char dato[30];
   cout<<“Ingrese nombre de producto\n";
   cin>>dato;

   cout<<”el dato en mayúscula es”<< strupr(dato);
}
```

strlwr

- Convierte las letras contenidas en una cadena a minúsculas

strlwr(cadena);

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

void main()
{
   char dato[30]="PROGRAMACION";

   cout<<"Convertido a minúsculas es "<< strlwr(dato);
}
```

strrev

- Invierte la cadena de caracteres

```
strrev(cadena);
```

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

void main()
{
   char dato[30]="PROGRAMACION";

   cout<<"Convertido a minúsculas es "<< strlwr(dato);
}
```