

como hasta ahora hemos realizado. Tenemos que utilizar obligatoriamente la función *strcpy*.

❑ FUNCIONES DE MANEJO DE CADENAS

✓ Estas funciones, que están en la librería '*string.h*' :

- No realizan reserva dinámica de memoria: Si una función necesita escribir una cadena en un parámetro, ese parámetro debe ser o un vector o un puntero a char inicializado con malloc o realloc.
- No comprueban los límites de los arrays, así que es responsabilidad del programador asegurar que los arrays o punteros que se le pasan a estas funciones son lo suficientemente grandes

✓ Las funciones más utilizadas son las siguientes:

- **char * strcat(char * cadena1, char * cadena2)**

Concatena (añade al final) una copia de cadena2 en cadena1 y añade al final de cadena1 el carácter nulo '\0'. El carácter nulo que originalmente tenía cadena1 se sustituye por el

primer carácter de cadena2. La cadena2 no se modifica en esta operación.

La función devuelve la (dirección de) cadena1 modificada por la concatenación.

Ejemplo: Programa que lee dos cadenas, pega la primera a la segunda y muestra el resultado por pantalla.

```
#include <stdio.h>
#include <string.h>
int main ()
{
    char cad1[30],cad2[30];
    gets(cad1);
    gets(cad2);
    strcat(cad2,cad1);
    puts(cad2);
}
```

- **int strcmp (const char*cad1, const char*cad2)**

Compara lexicográficamente, distinguiendo entre minúsculas y mayúsculas, dos cadenas que finalizan con el carácter nulo y devuelve un entero que se interpreta así:

Valor devuelto	Significado
$0 < \text{(negativo)}$	cad1 es menor que cad2
0	cad1 == cad2
$> 0 \text{ (positivo)}$	cad1 es mayor que cad2

Las funciones `stricmp` y `strcmipi`: ignoran la diferencia entre minúsculas y mayúsculas.

Ejemplo: Programa que verifica la introducción de contraseñas. La palabra clave es "paso".

```
#include <stdio.h>
#include <string.h>
int claveOK(); //Función booleana
int main ()
{
    if (claveOK()){
        printf("Palabra clave correcta");
        return(1);
    }
    else{
        printf("palabra clave invalida\n");
        return(0);
    }
}
int claveOK()
{
    char s[30];
    int nIntentos=3;
    do{
        printf("Introduzca una palabra clave: ");
```

```

    gets(s);
    if (strcmp(s,"paso"))
        nIntentos--; //Palabra clave no válida
    else
        break;      //Palabra clave válida
} while(nIntentos>0);

return(nIntentos);
}

```

• **char *strcpy (char *cad1, const char *cad2)**

Copia el contenido de cad2 en cad1, devolviendo cad1. Es lo mas parecido que tiene C a la operación de asignación de una cadena a una variable.

Ejemplo: Programa que copia "hola" en 'cad'

```

#include <stdio.h>
#include <string.h>
int main ()
{
    char cad[80];
    strcpy(cad,"hola");
    puts(cad);
}

```

• **int strlen (const char *cad1)**

Devuelve el número de caracteres de una cadena sin contar el '\0' final.

• **char *strlwr (char *cad1)**

Convierte cad1 a minúsculas.

• **char *strncat (char*cad1, const char*cad2, int n)**

Añade no más de n caracteres (un carácter nulo y los demás caracteres siguientes no son añadidos) de la cadena apuntada por cad2 al final de la cadena apuntada por cad1. El carácter inicial de cad2 sobrescribe el carácter nulo al final de cad1. El carácter nulo siempre es añadido al resultado.

Ejemplo: Programa que concatena dos cadenas e impide que se produzca desbordamiento

```
#include <stdio.h>
#include <string.h>
int main ()
{ char s1[5], s2[5];
  int lon;
  gets(s1); gets(s2);
  lon=4-strlen(s1);
  if(lon>=0) strncat(s1,s2,lon);
  puts(s1);
}
```

• **char* strncpy(char *dest, const char*orig, int n)**

Copia no más de n caracteres (caracteres posteriores al carácter nulo no son copiados) de la cadena 'orig' a la cadena 'dest'.

Ejemplo: Programa que copia toda una cadena leída por teclado en otra asegurándose de que no se produce desbordamiento.

```
#include <stdio.h>
#include <string.h>
int main ()
{   char s1[30], s2[5];
    gets(s1);
    strncpy(s2,s1,4);
    puts(s2);
}
```

- **char *strupr (char* cad)**
Convierte 'cad' a mayúsculas

□ FUNCIONES DE MANEJO DE CARACTERES

Estas funciones se aplican solo sobre caracteres y estan incluidas en las librerías '*ctype.h*' y '*stdlib.h*'.
(atof, atoi, atol, itoa)

- **double atof(const char *cad)**

Convierte una cadena en un double. Devuelve el double correspondiente y 0 si la cadena no es un numero.

- ***int atoi(char *cad)***

Convierte una cadena en un entero. Devuelve el entero correspondiente y 0 si la cadena no es un entero.

- ***long int atol(const char *cad)***

Convierte una cadena en un entero largo. Devuelve el entero correspondiente y 0 si la cadena no es un entero.

- ***int isalpha(int car)***

Devuelve un valor distinto de 0 (verdadero) si el carácter es una letra del alfabeto y 0 (falso) en otro caso.

- ***int isdigit(int car)***

Devuelve un valor distinto de 0 (verdadero) si el carácter es un dígito y 0 (falso) en otro caso.
0

- ***int isalnum(int car)***

Devuelve un valor distinto de 0 (verdadero) si el carácter es una letra o un dígito y 0 (falso) en otro caso. 0

- ***int islower(int car)***

Devuelve un valor distinto de 0 (verdadero) si el carácter es una letra en minúscula y 0 (falso) en otro caso. 0

- ***int ispunct(int car)***

Devuelve un valor distinto de 0 (verdadero) si el carácter es un símbolo (':','@, ...) y 0 (falso) en otro caso.

- ***int isspace(int car)***

Devuelve un valor distinto de 0 (verdadero) si el carácter es un espacio (' ',tabulación, intro,...) y 0 (falso) en cualquier otro caso.

- ***int isupper(int car)***

Devuelve un valor distinto de 0 (verdadero) si el carácter es una letra en mayúsculas y 0 (falso) en cualquier otro caso.

- ***int itoa(int entero, char *cad, int base)***

Convierte un entero a una cadena. La base especifica la base que debe ser usada en la conversión. Debe estar entre 2 y 36. Si el

entero es negativo y la base es 10 el primer carácter de la cadena será el signo menos (-).

- ***int tolower(int car)***

Devuelve el equivalente en minúscula del carácter. Si el carácter no es una letra del alfabeto no sufre ningún cambio.

- ***int toupper(int car)***

Devuelve el equivalente en mayúsculas del carácter. Si el carácter no es una letra del alfabeto no sufre ningún cambio.