

CURSOS
INTERSEMESTRALES



PROTECO

C Intermedio

Cadenas

¿Qué son las cadenas?

- Las cadenas de caracteres son declaradas en C como arreglos de caracteres y permiten la utilización de un cierto número de notaciones y de funciones especiales.



¿Qué es una cadena?

En C no existe un tipo predefinido para manipular cadenas de caracteres (string). Sin embargo, el estándar de C define algunas funciones de biblioteca para tratamiento de cadenas.

Una cadena en C es un arreglo de caracteres de una dimensión (vector de caracteres) que termina con el carácter especial '\0' (cero).

El formato para declarar una cadena es:

`char nombre[n];`

donde: $n \geq 1$ y representa a la longitud-1 real de la cadena.



¿Cómo declarar las cadenas?

En el caso especial de los arreglos de caracteres, podemos utilizar varias formas de inicialización:

```
char cadena[] = "Hola";
```

```
char cadena[] = {'H','o','l','a',0};
```

```
char cadena[] = {'H','o','l','a','\0'};
```

Sin especificar el tamaño de la cadena, o especificando el tamaño:

```
char cadena[5] = "Hola";
```

```
char cadena[5] = {'H','o','l','a',0};
```

```
char cadena[5] = {'H','o','l','a','\0'};
```



String.h

Es una biblioteca que contiene los prototipos de funciones de clasificación de caracteres.

Algunas de las funciones más utilizadas son las siguientes:

- strcpy
- strlen
- strcat
- strcmp
- strtok
- strstr



Nombres	Descripción
<code>memcpy</code>	copia n bytes entre dos áreas de memoria que no deben solaparse
<code>memmove</code>	copia n bytes entre dos áreas de memoria; al contrario que <code>memcpy</code> las áreas pueden solaparse
<code>memchr</code>	busca un valor a partir de una dirección de memoria dada y devuelve un puntero a la primera ocurrencia del valor buscado o <code>NULL</code> si no se encuentra
<code>memcmp</code>	compara los n primeros caracteres de dos áreas de memoria
<code>memset</code>	sobre escribe un área de memoria con un patrón de bytes dado
<code>strcat</code>	añade una cadena al final de otra
<code>strncat</code>	añade los n primeros caracteres de una cadena al final de otra
<code>strchr</code>	localiza un carácter en una cadena, buscando desde el principio
<code>strrchr</code>	localiza un carácter en una cadena, buscando desde el final
<code>strcmp</code>	compara dos cadenas alfabéticamente ('a'!= 'A')
<code>strncmp</code>	compara los n primeros caracteres de dos cadenas numéricamente ('a'!= 'A')
<code>strcoll</code>	compara dos cadenas según la colación actual ('a'== 'A')
<code>strcpy</code>	copia una cadena en otra
<code>strncpy</code>	copia los n primeros caracteres de una cadena en otra
<code>strerror</code>	devuelve la cadena con el mensaje de error correspondiente al número de error dado
<code>strlen</code>	devuelve la longitud de una cadena
<code>strspn</code>	devuelve la posición del primer carácter de una cadena que no coincide con ninguno de los caracteres de otra cadena dada
<code>strcspn</code>	devuelve la posición del primer carácter que coincide con alguno de los caracteres de otra cadena dada
<code>strpbrk</code>	encuentra la primera ocurrencia de alguno de los caracteres de una cadena dada en otra
<code>strstr</code>	busca una cadena dentro de otra
<code>strtok</code>	parte una cadena en una secuencia de tokens
<code>strxfrm</code>	transforma una cadena en su forma de colación (??)
<code>strrev</code>	invierte una cadena



`strcpy (c1, c2);` Copia c2 en c1

`strcat (c1, c2);` Añade c2 al final de c1

`int strlen (cadena);` Devuelve la longitud de la cadena

`int strcmp (c1, c2);` Devuelve cero si c1 es igual a c2

`Int strtok(cadena, caracteres);` Separa la cadena al encontrar un espacio o los caracteres especificaos.

`Int strstr(c1, c2);` localiza la primera aparición de la cadena apuntada por c1 de la secuencia de c2



Error de desbordamiento

¿Qué es?

El desbordamiento del búfer ocurre cuando un programa permite la entrada de escritura más allá del final del búfer asignado.

Esto se puede reparar mediante funciones en nuestra biblioteca string.h:

- strncpy
- strncmp
- strncat



Impresión con formato

Como sabemos, las funciones `printf` y `scanf` soportan una cadena de texto conteniendo códigos y banderas de formato para indicar diferentes tipos de dato y justificación.

Código	Formato
<code>%c</code>	un <code>char</code> (caracter)
<code>%d</code>	un entero con signo en notación de base decimal
<code>%i</code>	un entero con signo
<code>%e</code>	reales((pseudoreales como <code>double</code>)) en notación científica indicando el exponente con "e"
<code>%E</code>	reales((pseudoreales como <code>double</code>)) en notación científica indicando el exponente con "E"
<code>%f</code>	formato de punto flotante
<code>%g</code>	la opción más corta entre <code>"%e"</code> y <code>"%f"</code>
<code>%G</code>	la opción más corta entre <code>"%E"</code> y <code>"%F"</code>
<code>%o</code>	un entero sin signo en notación de base octal
<code>%s</code>	una cadena de caracteres
<code>%u</code>	un entero sin signo
<code>%x</code>	un entero sin signo en notación de base hexadecimal, usando minúsculas para los dígitos extendidos
<code>%X</code>	un entero sin signo en notación de base hexadecimal, usando mayúsculas para los dígitos extendidos
<code>%p</code>	un puntero
<code>%n</code>	un puntero a un entero en el cual se deposita la cantidad de caracteres escritos hasta el momento



Puts y gets

Estas funciones se encuentran en `stdio.h`, al igual que `printf` y `scanf`, imprimen y leen datos respectivamente, pero `puts` y `gets` lo hacen sin formato.

`puts()` recibe un único argumento que es la cadena de caracteres a mostrar en pantalla.

`gets()` recibe la variable en la que debe almacenarse una entrada del usuario que va a ser considerada como una cadena de caracteres.

