



FACULTAD DE INFORMÁTICA

Arrays y cadenas de caracteres

SUPLEMENTO

Cadenas de caracteres al estilo de C

Programación orientada a objetos — Unidad 4

Autor: Luis Hernández Yáñez

FdI
UCM

Cadenas de caracteres al estilo de C

Una cadena (de caracteres) es básicamente un array de caracteres.

En C++ se pueden utilizar las cadenas al estilo de C, aunque tiene definida una clase `string` mucho más versátil.

Cadenas de C:

`char nombre[tamaño];`

Arrays de caracteres de longitud variable (con tamaño máximo).

A diferencia de otros lenguajes, que mantienen un contador de caracteres, en C se detecta el final de la cadena mediante un carácter especial:

Cadenas terminadas en nulo

En la cadena

`char cad[80];`

realmente caben como máximo 79 caracteres; el nulo, un carácter con código ASCII 0 (`'\0'`), ocupa una posición del array.

FdI
UCM

Cadenas de caracteres al estilo de C

Las cadenas de caracteres se pueden inicializar:

```
char cad[25] = "Me gusta C++";
```

Se añade automáticamente el terminador nulo.

Se puede delegar en el compilador el cálculo de la longitud de una cadena que se inicializa (array de caracteres no delimitado):

```
char cad[] = "Me gusta C++";
```

Se añade automáticamente el terminador nulo y se establece la longitud en el número total de caracteres (incluyendo el nulo).

Las cadenas se pueden escribir en la pantalla (`cout`) y leer del teclado (`cin`), utilizando insertores y extractores de la misma forma como se hace con otros tipos de datos:

```
char cad[25];
```

```
...
```

```
cin >> cad;
```

```
cout << cad << endl;
```

Se puede usar
`const` para crear
cadenas constantes

FdI
UCM

Cadenas de caracteres al estilo de C

```
#include <iostream>
using namespace std;

int main()
{
    const int MAX = 80;
    char cad[MAX];

    cout << "Introduzca una cadena: ";
    cin >> cad;

    cout << "Cadena introducida: " << cad << endl;

    return 0;
}
```

Problema: No se comprueba si se sobrepasa el límite de un array.

Si se introducen más de 79 caracteres en el programa anterior, se estará usando memoria de forma ilícita.

Limitación del número de caracteres máximo a colocar en la cadena por medio del manipulador `setw()` (cabecera `iomanip`):

```
#include <iostream>
#include <iomanip>
using namespace std;

const int MAX = 10;
char cad[MAX];
...
cout << "Introduzca una cadena: ";
cin >> setw(MAX) >> cad;
// máximo de MAX caracteres, incluyendo el nulo
...
```

Problema de la lectura de cadenas con el extractor de `cin`:

Se termina la entrada con el primer espacio en blanco introducido. Se puede pasar un mensaje `getline` a `cin`:

`cadena a leer` → `cin.getline(cad, MAX);` → `máximo de caracteres a asignar (incluyendo el nulo)`

`getline()` admite un tercer argumento: carácter que hará que se termine la entrada (en lugar de `Intro`, que corresponde a `'\n'`).

Cuando se leen varias cadenas seguidas pueden aparecer problemas al leer la segunda y siguientes (no se asigna nada a las cadenas); se puede resolver el problema pasando un mensaje `ignore()` a `cin` después de cada lectura:

```
cin >> cad1;
cin.ignore();
cin >> cad2;
...
```

Funciones de manipulación de cadenas

En `cstring` `strlen()` `strcpy()` ...

Copia de cadenas

No se pueden copiar cadenas con el operador de asignación (`=`). Se deben copiar carácter a carácter:

```
#include <cstring>
...
for(int i = 0; i < strlen(cad1); i++)
    cad2[i] = cad1[i];
cad2[i] = '\0'; // carácter nulo de terminación
```

O utilizar funciones de manipulación:

```
#include <cstring>
...
strcpy(cad2, cad1);
```

Otras funciones de manipulación de cadenas

Requieren la inclusión de la cabecera `cstring`

<code>strlen(cad)</code>	Longitud de la cadena <code>cad</code> (sin contar el terminador nulo)
<code>strcpy(cad1, cad2)</code>	Copia la cadena <code>cad2</code> en <code>cad1</code>
<code>strcat(cad1, cad2)</code>	Agrega la cadena <code>cad2</code> al final de <code>cad1</code>
<code>strcmp(cad1, cad2)</code>	Compara las cadenas <code>cad1</code> y <code>cad2</code> 0 si las cadenas son iguales negativo si <code>cad1</code> es menor que <code>cad2</code> positivo si <code>cad1</code> es mayor que <code>cad2</code>