



SIRURI DE CARACTERE

O constanta de tip sir de caractere se declara intre caracterele ” “ (ghilimele). In memoria interna, o constanta de acest tip este retinuta sub forma unui vector de caractere. Fiecare componenta a sirului (incepand cu cea de indice 0) retine codul ASCII (American Standard Code for Information Interchange - Codul Standard American pentru Schimbul de Informatii) al caracterului pe care il memoreaza. Conventia este ca ultimul octet sa retina 0 (codul caracterului nul). Caracterul nul este memorat automat. Trebuie rezervate *lungimea_sirului+1* caractere char (+1 pentru caracterul nul).

Limbajul C/C++ permite initializarea unui tablou de caractere printr-o constanta sir, care include automat caracterul null.

Exemplu :

```
char vect[11]="calculator";  
char vect[]="calculator"; (compilerul face calculul numarului de octeti necesari)  
char vect[100]="calculator"; (s-au rezervat mai multi octeti decat era necesar)
```

Sirurile de caractere sunt de fapt *tablouri de caractere*, care au ca ultim element un terminator de sir, caracterul null.

**Exemplu:**

```
char tc[5] = {'a', 'b', 'c', 'd', 'e'}; // tablou de caractere
```

```
char sc[5] = {'a', 'b', 'c', 'd', '\0'}; // sir de caractere cu elementele abcd
```

Ultima initializare este echivalenta cu:

```
char sc[5] = "abcd"; //sau char sc[] = "abcd";
```

```
char sc1[5] = "abcd";
```

```
char s[10];
```

```
cout<<sc<<endl; //afiseaza abcd
```

```
cout<<tc<<endl; //eroare: tabloul de caractere nu contine terminatorul de sir, deci nu  
poate fi afisat ca sir
```

```
cout<<s<<endl; // eroare: tablou neinitializat
```

```
cout<<sc1[0]; // afiseaza primul caracter din sirul sc1
```

```
cout<<sc1[2]; // afiseaza al treilea element din sirul sc1
```

```
sc1[1]='K'; // elementului din sir de indice 1 i se atribuie valoarea 'K';
```