

# Șiruri de caractere în C++. Funcții specifice.

```
#include <cstring>
```

Șirurile de caractere sunt vectori de caractere, terminați cu `,\0'`.

## 1. Declaraire

```
char s[lgmax+1], voc[]="aeiouAEIOU", *p;
```

exemplu: `char s[101];` // se alocă 101 octeți pentru un șir de maxim 100 de caractere  
// s= **pointer constant** la primul element al șirului

```
char *p; // nu are memorie alocată în spatele pointerului dar poate referi în orice  
// moment o adresă din cadrul unui șir de caractere deja alocat
```

**POINTER** = variabilă care are ca valoare o adresă de memorie

## 2. Inițializare

a. Prin citire:

- Când șirul **NU** conține caractere albe (spațiu, tab, enter)

```
fin>>s; //citeste pana la intalnirea primului caracter alb  
cin>>s; //citeste pana la enter dar nu mai mult de lg_max car.
```

- Când șirul conține caractere albe

```
fin.getline(s, lg_max); //scoate și ENTER-ul din buffer  
fin.get(s, lg_max); //NU scoate și ENTER-ul din buffer,  
//urmatoarea citire va prelua doar ENTER
```

b. Prin copiere

```
char s[1001], t[1001];  
s=t; INCORECT! (deoarece s= pointer constant la începutul șirului)  
strcpy(s,t); CORECT! (copiază șirul t la adresa de început a șirului s,  
adica il suprascrive)
```

c. Prin atribuire

- la declarare

```
char voc[]="aeiouAEIOU", cif_pare[]="02468";
```

- prin utilizare pointer de tipul `char *p;`

```
p=s;
```

```
p=s+3; //șirul care începe cu al 4-lea caracter din s
```

**exemplu:** `char s[100]="exemplu de atribuire", *p;`

```
p=s+5;
```

```
cout<<p; //se va afișa: lu de atribuire
```

## 3. Afisarea unui șir de caractere

```
fout<<s;
```

```
cout<<s;
```

**Parcurgerea unui șir de caractere:**

```
int lg=strlen(s), i;  
for (i=0; i<lg; i++)  
    prel(s[i]);
```

**Prelucrarea vocalelor dintr-un text**

```
char sir[1001], voc[]="aeiouAEIOU";  
int i, lg;
```

```
...
```

```
lg=strlen(sir);
```

```
for (i=0; i<lg; i++)
```

```
    if (strchr(voc, sir[i])!=0)
```

```
        prel_vocala(sir[i]);
```

## FUNCȚII SPECIFICE

1. **strlen(s)**=lungimea efectivă a șirului (fara ,\0')
2. **Funcții de copiere** (atribuirea se realizează prin copiere)
  - a. **strcpy(sir\_dest, sir\_sursa);**
  - b. **strncpy(sir\_dest, sir\_sursa, nrcar);**  
sir\_dest[nrcar]='\0';  
(copiază în destinație primele nrcar din sursă)
3. **Funcții de concatenare**
  - a. **strcat(sir\_dest, sir\_sursa);**
  - b. **strncat(sir\_dest, sir\_sursa, nrcar);**  
sir\_dest[lg\_sir\_dest+nrcar]='\0';  
(concatenează la sfârșitul destinației primele nrcar din sursă)
4. **Funcții de comparare**
  - a. **strcmp(sir1, sir2);**- returnează o valoare negativă dacă sir1<sir2, o valoare strict pozitivă dacă sir1>sir2 și 0 dacă sunt egale
  - b. **stricmp(sir1, sir2);** - analog strcmp dar NU face distincție între litere mari și mici
5. **Funcții de căutare**
  - a. Căutare caracter: **strchr(sir, car)**- returnează pointer (adresă de memorie) la prima apariție a caracterului în șir sau NULL dacă nu există
  - b. Căutare subșir: **strstr(unde, pe\_cine)** - returnează pointer (adresă de memorie) la prima apariție a subșirului **pe\_cine** în șirul **unde** sau NULL dacă nu există
6. **Funcția de oglindire**  
**strrev(sir);** //oglindeste sirul de caractere transmis
7. **Funcții de conversie la majuscule/minuscule**
  - **strupr(sir);** //transforma sirul in litere mari
  - **strlwr(sir);** //transforma sirul in litere mici
  - **toupper(car);** //transforma caracterul in litera mare
  - **tolower(car);** //transforma caracterul in litera mica
8. **Funcții pe caractere (#include <ctype.h>)**
  - **isdigit(car)** //verifica daca e caracter cifra  
// s[i]>='0' && s[i]<='9'
  - **isalpha(car)** //verifica daca e caracter litera
  - **islower(car)** //verifica daca e caracter litera mica  
// s[i]>='a' && s[i]<='z'
  - **isupper(car)** //verifica daca e caracter litera mare  
// s[i]>='A' && s[i]<='Z'

## 9. Funcții de spargere a unui șir în entități delimitate de separatori precizați

**strtok(sir, sep)**=returneaza pointer la prima entitate delimitata; pentru urmatoarele se face apelul **p= strtok(NULL, sep)** ;

```
char sep[]=" ;.:,()?!-\"", sir[lgmax+1], *p;
p=strtok(sir,sep);
while (p)
    { prel_entitate(p);
      p=strtok(NULL, sep);
    }
```

Exemple:

- Stergerea vocalelor dintr-un șir s

```
i=0;
while (i<strlen(s))
    if (strchr("aeiouAEIOU", s[i])) strcpy(s+i, s+i+1);
    else i++;
```

- Numărarea aparițiilor unui cuvânt c în șirul s

```
char *p;
nr=0;
p=strstr(s, c);
while (p)
{
    nr++;
    p=strstr(p+1, c);
}
```