

ATENTIE! Atunci cand folosim functii ce prelucreaza siruri de caractere, noi transmitem de fapt un pointer catre acel sir. Am prezentat mai multe despre pointeri, in acest articol: [Pointeri in C++ | Introducere](#) . Asadar, atunci cand voi spune "functia primeste ca si parametru / parametrii un sir de caractere" voi sa va ganditi tot timpul ca aceasta primeste un pointer catre inceputul acestui sir.

Link articol: <https://tutoriale-pe.net/siruri-de-caractere-functii-predefinite-in-c>

1. STRLEN (String Length)

- **Parametrii:** sir de caracter
- **Returneaza:** lungimea unui sir

```
char s[128] = "ana";  
cout << "strlen(s) = " << strlen(s); // 3
```

2. STRCAT (String Concatenate)

- **Parametrii:** sir sursa, sir destinatie
- **NU returneaza nimic**

```
char s[128] = "merele sunt ";  
strcat(s, "sanatoase");  
cout << "strcat(s, \"sanatoase\") = " << s; // merele sunt sanatoase
```

3. STRCPY (String Copy)

- **Parametrii:** sir sursa, sir destinatie
- **NU returneaza nimic**

```
char s[128] = "mere";  
strcpy(s, "pere");  
cout << "strcpy(s, \"mere\") = " << s; // pere
```

4. STRCMP (String Compare)

- **Parametrii:** primul sir, al doilea sir
- **Returneaza:**
 - **< 0** - daca primul caracter diferit din s1 este mai mic decat cel din s2
 - **0** - **daca cele doua siruri sunt LA FEL**
 - **> 0** - daca primul caracter diferit din s1 este mai mare decat cel din s2



```
char s[128] = "mere";  
cout << "strcmp(s, \"mere\") = " << strcmp(s, "mere") << "\n"; // 0  
cout << "strcmp(s, \"pere\") = " << strcmp(s, "pere"); // -3
```

5. STRCHR (String Character)

- **Parametrii:** sir sursa, un caracter
- **Returneaza:** un pointer catre pozitia caracterului in sir

```
char s[128] = "Calul sare azi";  
char *p = strchr(s, 'u') ;  
cout << "strchr(s, 'u') = " << p; // ul sare azi
```

6. STRSTR (String String)

- **Parametrii:** sirul in care cauta, sirul ce este cautat
- **Returneaza:** un pointer catre pozitia sirului cautat in sirul original

```
char s[128] = "Calul sare azi";  
char *p = strstr(s, "sare") ;  
cout << "strstr(s, \"sare\") = " << p; // sare azi
```

7. STRTOK (String Token)

- **Parametrii:** sirul in care cauta, sir cu separatori
- **Returneaza:** un pointer catre inceputul primului token gasit

```
char s[128] = "Calul sare azi";  
char* cuvant = strtok(s, " ");  
cout << "strtok(s, \" \") = \n";  
while(cuvant != NULL) {  
    cout << "token: " << cuvant << "\n";  
    cuvant = strtok(NULL, " ");  
}
```

