

# ȘIRURI DE CARACTERE

## Funcții care lucrează cu șiruri de caractere (#include<cstring>)

### 1. Funcția strlen.

Are rolul de a returna lungimea unui șir (fără a lua în considerare caracterul nul).

Forma generală : **strlen**(nume sir);

**Exemplu:** Se citește un șir de caractere și se afișează numărul de caractere a șirului  
#include<iostream.h>

```
}
#include<string.h> void main()
{
    char s1[100];
    cin.get(s1,100);cin.get();
    cout<<"Șirul citit are"<<" "<<strlen(s1)<<" "<<"caractere";
```

### 2 Funcția strcpy.

Forma generală: **strcpy**(destinație,sursa)

Funcția are rolul de a **copia** șirul de la adresa **sursa** la adresa **destinație**.

```
#include<iostream.h>
#include<string.h> void main()
{
    char s1[20]="Limbalul Turbo C++",s2[20]="Limbajul C++";
    strcpy(s1,s2);
    cout<<s1;
}
```

### 3. Funcția strcat.

Forma generală: **strcat**(destinație,sursă)

Funcția are rolul de a **adăuga** șirului de la adresa **destinație** șirul de la adresa **sursă**.

```
#include<iostream.h>
#include<string.h>
void main()
{
    char s1[20]="Limbajul Turbo C++",s2[20]=" si Limbajul C++";
    strcat(s1,s2);
    cout<<s1;
}
```

### 4. Funcția strncat

Forma generală: **strncat**(destinație,sursă,n)

Funcția adaugă șirului **destinație** primii **n** octeți ai șirului **sursă**

```
#include<iostream.h>
#include<string.h>
void main()
{
    char s1[20]="Limbajul Turbo C++",s2[20]=" și Limbajul C++";
    strncat(s1,s2,5);
    cout<<s1;
}
```

## 5. Funcția strchr

Forma generală: **strchr**(nume șir, 'caracter' )

Funcția caută în șirul **nume șir** caracterul **caracter** și returnează subșirul care începe cu prima apariție a caracterului citit

```
#include<iostream.h>
#include<string.h>
void main()
{
    char s1[20]="Limbajul Turbo C++";
    cout<<strchr(s1 , 'T');
}
```

## 6. Funcția strrchr

Returnează adresa ultimei apariții a caracterului căutat **strrchr**(șir, 'caracter' )

```
char s1[20]="Limbajul Turbo C++";
cout<<strrchr(s1 , 'u')-s1;
```

Returnează valoarea **10**

## 7. Funcția strcmp

Forma generală **strcmp**(sir1, sir2 )

Funcția are rolul de a compara 2 șiruri de caractere și va returna valoarea: **< 0**

dacă **sir1<sir2**

**= 0** dacă **sir1=sir2 > 0** dacă **sir1>sir2**

**Exemplu** Se compară șirul **a** cu șirul **b** și se listează relația dintre cele 2 șiruri astfel:

- dacă primele **n** caractere sunt identice se compară caracterele **n+1**
- dacă caracterul **n+1** al șirului **a** este situat alfabetic înaintea caracterului **n+1** al șirului **b** se afișează **a<b**
- dacă caracterul **n+1** al șirului **a** este situat alfabetic după caracterul **n+1** al șirului **b** se afișează **a>b**
- dacă primul caracter al șirului **a** este situat alfabetic înaintea primului caracter al șirului **b** se afișează **a<b** indiferent de lungimea celor 2 șiruri
- dacă primul caracter al șirului **a** este situat alfabetic după primul caracter al șirului **b** se afișează **a>b** indiferent de lungimea celor 2 șiruri

```
#include<iostream.h>
```

```
#include<string.h>
```

```
void main()
```

```
{
    char a[20], b[20];
    int semn;
    cout<<"Introduceti sirul a: "; cin>>a;
    cout<<"Introduceti sirul b: "; cin>>b;
    semn=strcmp(a,b); if(semn<0)
    cout<<"a < b"; else
```

```
}
```

```
if(semn>0) cout<<"a > b";
else cout<<"a = b";
```

## 8. Funcțiile strlwr și struwr

**strlwr(s1)** - convertește toate literele șirului si în **litere mici**

**struwr(s2)** - convertește toate literele șirului s2 în **litere mari**

```
#include<iostream.h>
```

```
#include<string.h>
```

```
void main()
```

```
{
```

```
    char a[100]="este acesta un sir?", b[100]="Acest Lucru Nu Ma Deranjeaza"; cout<<struwr(a)<<endl<<strlwr(b);
```

```
}
```

## 9. Funcția strstr

Forma generală: **strstr(sir1 ,sir2)**

Funcția identifică dacă șirul **sir2** este subșir al șirului **sir1**

## 10. Funcția strchr

Forma generală: **strchr(s1,s2)**

Funcția returnează **numărul caracterelor** din șirul **si** care **nu se găsesc** în șirul **s2**

## 11. Funcția strspn

Forma generală: **strspn(s1,s2)**

Funcția returnează **numărul caracterelor** din șirul **s1** care **se găsesc** în șirul **s2**

## 12. FUNCȚII UTILIZATE PENTRU CONVERSIA VALORILOR NUMERICE în SIR

La utilizarea acestor funcții se introduce directiva **#include<cstdlib>**

a) **Funcția atof** - convertește un șir către tipul **double**

b) **Funcția atold** - convertește un șir către tipul **long double**

```
#include <stdlib.h>
```

```
#include <iostream.h>
```

```
void main()
```

```
{
```

```
    float f;
```

```
    char *str = "12345.67"; f = atof(str);
```

```
    cout<<"string = "<<str<<endl<<"float = "<< f;
```

```
}
```

c) **Funcția atoi** - convertește un șir către tipul **int**

d) **Funcția atol** - convertește un șir către tipul **long**

**Exemplu** #include

```
<stdlib.h>
```

```
#include <iostream.h>
```

```
void main()
```

```
{
```

```
    int n;
```

```
    char *str = "12345.67"; n = atoi(str);
```

```
    cout<<"string = "<<str<<endl<<"float = "<< n;
```

```
}
```

e) **Funcția ecvt** - convertește o **valoare double** către un **sir**

f) **Funcția itoa** - convertește o **valoare de tip int** către un **sir**

g) **Funcția ltoa** - convertește o **valoare de tip long int** către un **sir**