



FUNCTII PENTRU OPERATII CU SIRURI DE CARACTERE

Funcțiile pentru operații cu siruri se găsesc în header-ul **<string.h>**.

1. Funcția **strlen**

strlen(nume_sir); – returnează lungimea efectivă a unui sir (fără a număra terminatorul de sir).

Exemplul 1:

```
#include<iostream>
#include<string.h>
using namespace std;
char a[50]="Suntem la ora de informatica";

int main()
{
cout<<"Lungimea sirului este: "<<strlen(a);
}
```

Programul va returna

Lungimea sirului este: 28

(24 de caractere + 4 spații)



1. Funcția strcpy

strcpy(sir_destinatie,sir_sursa); – copiaza sirul sir_ sursa in sir_destinatie (se simuleaza atribuirea a=b).

ATENȚIE!! Nu este permisa atribuirea intre doua siruri de caractere folosind operatorul =:Atribuirea se face folosind functia strcpy.

Exemplul 2:

```
#include<iostream>
#include<string.h>
using namespace std;

char a[50]="Afara ploua", b[40]="Este soare ";

int main()
{
    cout<<"Sirul a este:"<<a<<endl;
    strcpy(a,b);
    cout<<"Sirul a este:"<<a<<endl;
    cout<<"Sirul b este:"<<b;
}
```

Programul va returna

Sirul a este: Afara ploua (inainte de strcpy)

Sirul a este: Este soare

Sirul b este: Este soare



1. Funcția strcat

strcat(destinatie,sursa); – adauga sirului destinatie sirul sursa. Sirul sursa ramane nemodificat. Operatia se numeste **concatenare** si nu este comutativa.

Exemplul 3:

```
#include<iostream>
#include<string.h>
using namespace std;

char a[40]="Afara ploua";
char b[30]=" si este frig ";

int main()
{
    cout<<"Sirul a este:"<<a<<endl;
    cout<<"Sirul b este:"<<b<<endl;
    cout<<"Sirul a este:"<<strcat(a,b);
}
```

Programul va returna

Sirul a este: Afara ploua (inainte de strcat)
Sirul a este: si este frig
Sirul a este: Afara ploua si este frig



1. Funcția **strncat**

strncat(destinatie, sursa, nr); – adauga sirului destinatie primele nr caractere din sirul sursa. Sirul sursa ramane nemodificat.

Exemplul 4:

```
#include<iostream>
#include<string.h>
using namespace std;

char a[40]="Vine ";
char b[30]="vacanta? ";

int main()
{
    cout<<"Sirul a este:"<<a<<endl;
    cout<<"Sirul b este:"<<b<<endl;
    cout<<"Sirul a este:"<<strncat(a,b,4);
}
```

Programul va returna

**Sirul a este: Vine
Sirul a este: vacanta?
Sirul a este: Vine vaca**



5. Functia strchr

strchr(sir,c); – cauta caracterul **c** in sirul **sir**. Cautarea se face de la stanga la dreapta, iar functia intoarce adresa subsirului care incepe cu prima aparitie a caracterului **c**. Daca nu este gasit caracterul, functia returneaza 0. **Diferenta dintre adresa sirului initial si cea a subsirului returnat reprezinta chiar pozitia caracterului cautat in sirul dat.**

Exemplul nr. 5.1.

```
#include<iostream>
#include<string.h>
using namespace std;
int main()
{
    char a[100], p, c;
    cout<<"Dati sirul:";
    cin.get(a,100);
    cout<<"Dati caracterul cautat:";
    cin>>c;
    cout<<strchr(a,c)<<endl;
    cout<<"Dati caracterul cautat:";
    cin>>p;
    cout<<strchr(a,p);
}
```

Pentru sirul a= Ana are mere si mure

c=m

p=x

programul va returna:

mure si mere

0



Exemplul 5.2: Sa se afiseze toate pozitiile unui caracter intr-un sir

```
#include<iostream>
#include<string.h>
using namespace std;
int main()
{
    char a[100],*p, c;
    cout<<"Dati sirul:";
    cin.get(a,100);
    cout<<"Dati caracterul cautat:";
    cin>>c;
    p=strchr(a,c);
    while (p)
    {
        cout<<"Pozitia caracterului cautat este:"<<p-a<<endl;
        p++;
        p=strchr(p,c);
    }
}
```

Pentru sirul a= Ana are mere si mure si c=m programul va returna:

Pozitia caracterului cautat este:8
Pozitia caracterului cautat este:16

OBS:
Numararea incepe de la zero.
Se numara si spatiile

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
a	n	a		a	r	e		m	e	r	e		s	i		m	u	r	e

a=0
p= 8
p-a=8-0=8

a=0
p=16
p-a=16-0=16



6. Functia strrchr

strrchr(sir,c); – are același rol cu **strchr**, cu deosebirea ca returnează adresa ultimei apariții a caracterului (cautarea se face de la dreapta spre stanga; **r = right**)

Exemplul 6.1:

```
#include<iostream>
#include<string.h>
using namespace std;
int main()
{
    char a[100], p, c;
    cout<<"Dati sirul:";
    cin.get(a,100);
    cout<<"Dati caracterul cautat:";
    cin>>c;
    cout<<strrchr(a,c)<<endl;
    cout<<"Dati caracterul cautat:";
    cin>>p;
    cout<<strrchr(a,p);
}
```

Pentru sirul a= Ana are mere si mure

c=m

p=r

programul va returna:

mure

re



Exemplul 6.2: Sa se afiseze ultima pozitie a unui caracter, citit de la tastatura, intr-un sir

```
#include<iostream>
#include<string.h>
using namespace std;
int main()
{
    char a[100],*p, c;
    cout<<"Dati sirul:";
    cin.get(a,100);
    cout<<"Dati caracterul cautat:";
    cin>>c;
    p=strrchr(a,c);
    while (p)
    {
        cout<<"Pozitia caracterului cautat este:"<<p-a<<endl;
        p++;
        p=strrchr(p,c);
    }
}
```

Pentru sirul a= Ana are mere si mure si c=m programul va returna:

Pozitia caracterului cautat este:16



7. Functia **strcmp**

int **strcmp**(sir1, sir2); – are rolul de a compara doua siruri de caractere.

Valoarea returnata este <0 (daca sir1<sir2)
=0 (daca sir1=sir2)
>0 (daca sir1>sir2)

Functia strcmp face distinctie intre literele mari si literele mici ale alfabetului.

Observatie:

Functia **strcmp** returneaza diferenta dintre codurile ASCII ale primelor caractere care nu coincid.



Exemplul 7: Sa se compare doua siruri citite de la tastatura tinand cont de literele mici si mari

```
#include<iostream>
#include<string.h>
using namespace std;

char s1[100], s2[100];
int main()
{
    int x;
    cout<<"Dati sirul 1:";
    cin.get(s1,100);
    cin.get();
    cout<<"Dati sirul 2:";
    cin.get(s2,100);
    x=strcmp(s1,s2);
    if(x>0) cout<<"Primul sir este mai mare ";
    else if(x==0) cout<<"Sirurile sunt egale";
    else cout<<"Al doilea sir este mai mare decat
    primul ";
}
```

Programul va returna

Dati sirul 1: ANA

Dati sirul 2: ana

Al doilea sir este mai mare decat primul

(Codurile ASCII A=65 a=97)



8. Funcția **stricmp**

int stricmp(sir1,sir2); – are același rol cu **strcmp**, cu deosebirea că nu face distincție între literele mari și cele mici ale alfabetului (**i = ignore**).

Exemplul 8:

Sa se compare doua siruri citite de la tastatura fara a tine cont de literele mici si mari

```
#include<iostream>
#include<string.h>
using namespace std;
char s1[100], s2[100];
int main()
{
    int x;
    cout<<"Dati sirul 1:";
    cin.get(s1,100);
    cin.get();
    cout<<"Dati sirul 2:";
    cin.get(s2,100);
    x=stricmp(s1,s2);
    if(x>0) cout<<"Primul sir este mai mare ";
    else if(x==0) cout<<"Sirurile sunt egale";
    else cout<<"Al doilea sir este mai mare decat primul ";
}
```

Programul va returna

Dati sirul 1: ANA
Dati sirul 2: ana
Sirurile sunt egale



9. Funcția **strstr**

strstr(sir1,sir2); – are rolul de a identifica daca sirul **sir2** este subsir al sirului **sir1**.

Daca este, functia returneaza adresa de inceput a subsirului sir2 in sirul sir1, altfel returneaza adresa 0. In cazul in care sir2 apare de mai multe ori in sir1, se returneaza adresa de inceput a primei aparitii. Cautarea se face de la stanga la dreapta.

```
#include<iostream>
#include<string.h>
using namespace std;
char s1[100], s2[100],*x;
int main()
{
    cout<<"Dati sirul 1:";
    cin.get(s1,100);
    cin.get();
    cout<<"Dati sirul 2:";
    cin.get(s2,100);
    x=strstr(s1,s2);
    cout<<x;
}
```

Dati sirul 1: Ana are mere
Dati sirul 2: are are mere

```
#include<iostream>
#include<string.h>
using namespace std;
char s1[100], s2[100], x;
int main()
{
    cout<<"Dati sirul 1:";
    cin.get(s1,100);
    cin.get();
    cout<<"Dati sirul 2:";
    cin.get(s2,100);
    cout<<strstr(s1,s2);
}
```

Dati sirul 1: Ana are mere si pere
Dati sirul 2: mere mere si pere



10. Functia **strspn**

`int strspn(sir1,sir2);` – are rolul de a returna numarul de caractere ale sirului **sir1** (caractere consecutive care incep obligatoriu cu primul caracter) care se gasesc in sirul **sir2**.

Exemplu:

```
#include<iostream>
#include<string.h>
using namespace std;
char s1[100], s2[100], x;
int main()
{
    cout<<"Dati sirul 1:";
    cin.get(s1,100);
    cin.get();
    cout<<"Dati sirul 2:";
    cin.get(s2,100);
    cout<<strspn(s1,s2);
}
```



11. Functia **strcspn**

int strcspn(sir1,sir2); – are rolul de a returna numarul de caractere ale sirului **sir1** (caractere consecutive care incep obligatoriu cu primul caracter) care nu se gasesc in sirul **sir2**.

12. Functia **strlwr**

strlwr(sir); – are rolul de a converti toate literele mari din sir in litere mic. Restul caracterelor raman neschimbate.

13. Functia **strupr**

strupr(sir); – are rolul de a converti toate literele mici din sir in litere mari.

Restul caracterelor raman neschimbate



14. Functia **strbrk**

strpbrk(sir1,sir2); – actioneaza in felul urmator:

- Cauta primul caracter al sirului sir1 in sir2. Daca este gasit, returneaza adresa sa din cadrul sirului sir1 si executia se termina. Altfel, se trece la pasul urmator.
- Cauta al doilea caracter al sirului sir1 in sir2. Daca este gasit, returneaza adresa sa din cadrul sirului sir1 si executia se termina. Altfel, se trece la pasul urmator.
- ...

Daca nici un caracter al sirului sir1 nu apartine sirului sir2, functia returneaza adresa nula.



15. Functia **atof**

double atof(sir); – convertește un sir către tipul double. Dacă această conversie esuează (se întâlnește un caracter nenumeric), valoarea întoarsă este 0. Această funcție (ca și cele similare) necesită includerea bibliotecii **stdlib.h**.

16. Functia **atold**

long double atold(sir); – convertește un sir către tipul long double. Dacă această conversie esuează, valoarea întoarsă este 0.

17. Functia **atoi**

int atoi(sir); – convertește un sir către tipul int. Dacă această conversie esuează (se întâlnește un caracter nenumeric), valoarea întoarsă este 0.



18. Functia **atol**

long atol(sir); – converteste un sir catre tipul long. Daca aceasta conversie esueaza (se intalneste un caracter nenumeric), valoarea intoarsa este 0.

19. Functia **itoa**

itoa(int valoare,sir,int baza); – converteste o valoare de tip int in sir, care este memorat in variabila sir. Baza retine baza de numeratie catre care sa se faca conversia. In cazul bazei 10, sirul retine si eventualul semn -.

20. Functia **ltoa**

ltoa(long valoare,sir,int baza); – converteste o valoare de tip long int in sir, care este memorat in variabila sir.

21. Functia **ultoa**

ultoa(unsigned long valoare, sir, int baza); – converteste o valoare de tip unsigned long in sir, care este memorat in variabila sir.