

Funcții pe siruri de caractere

Funcția strlen: *int strlen(ume_sir);* – returnează lungimea efectivă a unui sir (fără a număra terminatorul de sir).

Exemplu: char a[50]="ora de informatica";
strlen(a) = 18

Funcția strcpy: *strcpy(sir_destinatie,sir_sursa);* – copiază sirul sir_sursa în sir_destinatie (se simulează atribuirea a=b înșă

ATENȚIE: Nu este permisă atribuirea între două siruri de caractere folosind operatorul =. Atribuirea se face obligatoriu cu funcția strcpy.

Exemplu:

char a[50]="primul sir",b[40]="al doilea sir";
strcpy(a,b);
a = "al doilea sir"; b="al doilea sir";

Funcția strcat: *strcat(dest,sursa);* – adaugă sirului dest sirul sursa.

Sirul sursa rămâne nemodificat. Operația se numește concatenare și nu este comutativă.

Exemplu:

char *a="Colegiul ",*b="Titeica?";
strcat(a,b);
a = "Colegiul Titeica?";

Funcția strncat: *strncat(dest,sursa,nr);* – adaugă dest primele nr caractere din sirul sursa. Sirul sursa rămâne nemodificat.

Exemplu:

char *a="Colegiul",*b="Titeica?";
strncat(a,b,3);
a = "Colegiul Tit";

Funcția strchr: *strchr(sir,c);* – are rolul de a căuta caracterul c în sirul sir.

Căutarea se face de la stânga la dreapta, iar funcția întoarce adresa subsirului care începe cu prima apariție a caracterului c. Dacă nu este găsit caracterul, funcția

returneaza 0. Diferenta dintre adresa sirului initial si cea a subsirului returnat reprezinta chiar pozitia caracterului cautat in sirul dat.

Exemplu:

```
char *a="acesta este un sir",b='t',c='x',d;  
cout<<strchr(a,b); se tipareste "ta este un sir";  
cout<<strchr(a,c); nu se tipareste nimic (se tipareste 0 daca se face o conversie  
la int a lui strchr(a,c) ;  
d= strchr(a,b);  
cout<<"Caracterul apare prima data la pozitia "<<d-a;
```

Sa se afiseze toate pozitiile unui caracter intr-un sir

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

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

```
void main()
```

```
{char a[100],*p,c;
```

```
cin.get(a,100);
```

```
cin>>c;
```

```
p=strchr(a,c);
```

```
while (p)
```

```
{cout<<"Pozitia "<<p-a<<endl;
```

```
p++;
```

```
p=strchr(p,c);}
```

```
}
```

Functia strrchr: *strrchr(sir,c);* – are acelasi rol cu strchr, cu deosebirea ca returneaza adresa ultimei aparitii a caracterului.

Cautarea se face de la dreapta spre stanga; r = right)

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).

Observatii:

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

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

Functia stricmp: *int stricmp(sir1,sir2);* – are acelasi rol cu strcmp, cu deosebirea ca nu face distinctie intre literele mari si cele mici ale alfabetului (i = ignore).

Functia 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

Functia strtok: *strtok(sir1,sir2);* – are rolul de a separa sirul sir1 in mai multe siruri (cuvinte) separate intre ele prin unul sau mai multe caractere cu rol de separator. Sirul sir2 este alcatuit din unul sau mai multe caractere cu rol de separator.

Functia strtok actioneaza in felul urmator:

- o Primul apel trebuie sa fie de forma strtok(sir1,sir2); Functia intoarce adresa primului caracter al primei entitati. Dupa prima entitate, separatorul este inlocuit automat prin caracterul nul.
- o Urmatoarele apeluri sunt de forma strtok(NULL,sir2); De fiecare data, functia intoarce adresa de inceput a urmatoarei entitati, adaugand automat dupa ea caracterul nul.
- o Cand sirul nu mai contine entitati, functia returneaza adresa nula.

Exemplu: Se citeste de la tastatura un sir de caractere. Sa se afiseze cuvintele din sir. Cuvintele sunt despartite de spatiu.

```
#include <iostream.h>

#include <conio.h>

#include <string.h>

void main()

{char a[100],cuv[10][10],*p,separator[]=" ";

int i=0,nr=0;

cout<<"Dati sirul:";cin.get(a,100);

strcpy(p,a);

p=strtok(p,separator);

while (p)

    {strcpy(cuv[++nr],p);

    p=strtok(NULL,separator);}

cout<<"Sunt "<<nr<<" cuvinte:"<<endl;

for (i=1;i<=nr;i++) cout<<cuv[i]<<endl;
```

}

Functii de conversie

Functia strlwr() cu forma generala

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

Functia strupr() cu forma generala

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

Functia strbrk() cu forma generala

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

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

Functia atof() cu forma generala

double atof(sir); – converteste un sir catre tipul double. Daca aceasta conversie esueaza (se intalneste un caracter nenumeric), valoarea intoarsa este 0. Aceasta functie (ca si cele similare) necesita includerea librariei stdlib.h.

Functia atold() cu forma generala

long double _atold(sir); – converteste un sir catre tipul long double. Daca aceasta conversie esueaza, valoarea intoarsa este 0.

Functia atoi() cu forma generala

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

Functia atol() cu forma generala

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

Functia itoa() cu forma generala

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 -.

Functia ltoa() cu forma generala

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

Functia ultoa() cu forma generala

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

```
#include<iostream.h>
#include<string.h>
#include<stdlib.h>
void main()
{char sir[100]; int n;
cout<<"dati sirul ";cin.get(sir,100);
n=atoi(sir);
cout<<"numarul dupa conversie "<<n<<"\n";
itoa(n,sir,10);
cout<<n<<" in baza 10 "<<sir<<"\n";
itoa(n,sir,8);
cout<<n<<" in baza 8 "<<sir<<"\n";
itoa(n,sir,16);
cout<<n<<" in baza 16 "<<sir<<"\n";
```

```
itoa(n,sir,2);  
cout<<n<<" in baza 2 "<<sir<<"\n";  
}
```

Sursa:

1. Citește mai mult: <http://infoliceu.webnode.com/clasa-a-xi-a-c/siruri-de-caractere/functii-de-conversie/>
2. Citește mai mult: <http://infoliceu.webnode.com/clasa-a-xi-a-c/siruri-de-caractere/functii-pentru-siruri-de-caractere/>

Creați propriul site gratuit: <http://www.webnode.ro>