

Șir de caractere

Limbajul C nu dispune de un tip de date nativ pentru reprezentarea șirurilor de caractere de lungime variabilă. În acest scop se utilizează structuri de date de tip tablou de caractere. Întrucât șirurile de caractere prelucrate în programe au în general lungime variabilă, s-a stabilit o convenție prin care ultimul caracter utilizat al unui șir este urmat de un caracter cu valoarea zero ('\0'), numit terminator de șir.

```
char sir [10];
```

Exemplul de mai sus declară un tablou de 10 de elemente de tip caracter. Un asemenea tablou se poate folosi pentru memorarea unui șir de caractere de lungime variabilă, dar de maxim 9 de caractere, întrucât ultimul element este rezervat pentru terminatorul de șir.

Compilatoarele de C permit inițializarea tablourilor de caractere în momentul declarării acestora cu un șir de caractere:

```
char sir [12] = "Input text";
```

În urma acestei declarații, variabila *sir* va avea urmatorul conținut:

Index	0	1	2	3	4	5	6	7	8	9	10	11
Conținut	I	n	p	u	t		t	e	x	t	\0	-

Principalele funcții din biblioteca standard C pentru lucrul cu șiruri de caractere

Întrucât nu există tip de date nativ șir de caractere, limbajul nu dispune nici de operatori pentru prelucrarea șirurilor. Există însă o serie de funcții care asistă programatorul în lucrul cu șirurile de caractere. Aceste funcții necesită includerea fișierului header **string.h**. Toate funcțiile din această bibliotecă presupun că variabilele de tip șir de caractere asupra cărora operează respectă convenția de a avea un terminator (caracter cu codul 0).

Citirea și afisarea unui șir de caractere de la tastatură

```
char sir [50];
printf ("Introduceti un sir: ");
scanf ("%s", sir); //va citi până la primul caracter alb (spațiu/tab/enter)
printf ("Ati tastat: %s \n", sir);
```

Se recomandă utilizarea funcției *gets*:

```
char sir [50];
printf ("Introduceti un sir: ");
gets (sir);
printf ("Ati tastat: %s \n", sir);
```

Lungimea unui șir de caractere

```
char sir [80];
printf ("Introduceti un sir: ");
gets (sir);
int len = strlen (sir);
printf ("Lungimea sirului este: %d \n", len);
```

Copierea conținutului unui șir de caractere

```
char sir1[80], sir2[80];
printf ("Introduceti un sir: ");
gets (sir1);
strcpy (sir2, sir1); // sir2 - destinatie, sir1 - sursa
/* nu este permisa atribuirea sir2 = sir1 ! */
printf ("Ati tastat: %s \n", sir2);
```

Compararea alfabetica a doua siruri de caractere

```
char sir1 [80], sir2 [80];
int x;
printf ("Introduceti primul sir: ");
gets (sir1);
printf ("Introduceti al doilea sir: ");
gets (sir2);
x = strcmp (sir1, sir2);
```

```

if (x > 0)
printf ("%s > %s \n", sir1, sir2);
else
if (x == 0)
printf ("%s == %s \n", sir1, sir2);
else
printf ("%s < %s \n", sir1, sir2);

```

Cautarea unui subsir într-un sir

```

char sir [80], subsir [80];
char *p;
printf ("Introduceti sirul: ");
gets (sir1);
printf ("Introduceti subsirul cautat: ");
gets (sir2);
p = strstr (sir, subsir);
if (p != NULL)
printf ("%s contine %s \n", sir, subsir);
else
printf ("%s nu contine %s \n", sir, subsir);

```

Concatenarea a doua siruri de caractere

```

char sir1[40], sir2[40], sir [80]= "";
printf ("Introduceti primul sir: ");
gets (sir1);
printf ("Introduceti al doilea sir: ");
gets (sir2);
strcpy (sir, sir1);
strcat (sir, sir2);
printf ("Ati introdus: %s \n", sir);

```

Probleme rezolvate

Exemplu: Programul următor citește de la tastatură un nume și determină dacă este nume de fată sau de băiat (toate numele românești de fată, cu excepția numelui "Carmen" se termină cu o vocală):

```

#include <stdio.h>
#include <string.h>
#include <ctype.h> /* pentru funcția toupper */
int main ()
{
    char nume [80];
    int n;
    char c;
    do {
        printf ("Introduceti numele: ");
        gets (nume);
        n = strlen (nume);
    }while (n == 0);
    /* nu permitem introducerea unui sir vid */
    c = toupper (nume [n - 1]);
    if ((c=='A')||(c=='E') || (c=='I') || (c=='O') || (c=='U'))
        printf ("Nume de fata \n");
    else
        printf ("Nume de baiat \n");
return 0;
}

```

Exemplu: Următorul program citește de la tastatură un șir de caractere și afișează șirul în ordine inversă:

```

#include <stdio.h>
#include <string.h>

```

```

int main (){
    char sir [80];
    int i, n;
    printf ("Introduceti sirul: ");
    gets (sir);
    n = strlen (sir);
    for (i = n - 1; i >= 0; i--)
        printf ("%c", sir [i]);
    printf ("\n");
    return 0;
}

```

Exemplu:

Sa se scrie un program C care realizează următoarele:

- Șterge dintr-un șir de caractere un subșir specificat prin poziție și lungime.
- Inserează într-un șir începând cu o poziție dată un alt șir.

```

#include <stdio.h>
#include <string.h>
int main()
{
    char sir1[50]="";
    int nr, poz1, poz2, i=0;
    printf ("Introduceti un sirul 1:");
    scanf("%s", sir1);
    //gets(sir1);
    //fgets(sir1, 50, stdin);
    /*do {
        sir1[i] = getchar();
        i++;
    } while (sir1[i-1] != '\n');
    sir1[i-1] = '\0';*/
    printf ("Ati tastat 1: %s \n", sir1);
    printf ("Introduceti numarul de caractere ce trebuie de sters: ");
    scanf("%i", &nr);
    printf ("Introduceti pozitia de unde incepe subsirul ce trebuie de sters: ");
    scanf("%i", &poz1);

    strcpy (sir1+poz1, sir1+poz1+nr);    // unu - destinatie, doi - sursa
    printf ("Sirul modificat 1: %s \n", sir1);

    //Insereaza într-un sir începând cu o pozitie data un al sir
    char sir2[10]="", aux[50]="";
    printf ("Introduceti un sirul de introdus:");
    scanf("%s", sir2);
    //printf ("Ati tastat 2: %s \n", sir2);
    puts(sir2);
    printf ("Introduceti pozitia de unde incepe inserarea subsirului: ");
    scanf("%i", &poz2);
    strcpy(aux, sir1 + poz2);
    strcpy(sir1 + poz2, sir2);
    strcat(sir1, aux);
    puts(sir1);
    return 0;
}

```

Exemplu:

Să se scrie un program C care să citească de la tastatură un cuvânt și să verifice dacă respectivul cuvânt este palindrom (cuvânt care poate fi citit de la stânga la dreapta și de la dreapta la stânga fără să-și piardă sensul: cojoc, capac, rar).

```

#include <stdio.h>
#include <string.h>
int main (){
    char str [50], strInvert[50] = "";
    int i, n, index = 0;
    printf ("Introduceti sirul: ");
    gets (str);
    n = strlen (str);
    for (i = n - 1; i >= 0; i--) {
        strInvert[index] = str[i];
        index++;
    }
    strInvert[index] = '\0';
    puts(strInvert);
    printf("len = %i\n", strlen(strInvert));
    if (strcmp(str, strInvert) == 0)
        printf("\"%s\" este palindrom", str);
    else printf("\"%s\" nu este palindrom", str);
return 0;
}

```