

Siruri de caractere în limbajul C

BREVIAR TEORETIC

SIRURI DE CARACTERE. DECLARARE

Pentru memorarea șirurilor de caractere, în C, se folosesc tablourile unidimensionale de caractere. Declarația, în acest caz, se face conform sintaxei:

```
char id_sir[ nr_elem];
```

Pentru marcarea sfârșitului de șir, după ultimul caracter se adaugă un octet cu valoarea 0. Pentru a sugera că este vorba de un caracter, se folosește secvența '\0'. În cazul particular al folosirii unui șir vid, primul element din tablou este chiar indicatorul de sfârșit de șir:

```
sir[ 0 ] = 0; /* sau sir[0] = '\0'; */
```

Deci, un tablou de *nr_elem* caractere poate memora șiruri cu lungimi cuprinse între 0 și *nr_elem* - 1 caractere, deoarece trebuie să "încapă" și indicatorul de sfârșit de șir '\0' în zona de memorie rezervată tabloului.

Exemplu:

```
int n = 0;
char my_str[] = "Un test";
while( my_str[ n++ ] ) ; //while (my_string[n++] );
printf( "Lungimea este: %d\n", n-1 );
```

SIRURI DE CARACTERE. INITIALIZARE

O succesiune de caractere încadrată între ghilimele reprezintă o constantă șir de caractere (ghilimelele nu fac parte din șirul de caractere).

La întâlnirea constantei, compilatorul rezervă zona de memorie necesară, o inițializează cu codurile ASCII ale șirului și adaugă automat, după acestea, indicatorul de sfârșit de șir '\0'

Exemplu

```
char s[] = { 'A', 'B', 'C', 'D', 'E', '\0' };

char s1[] = "ABCDE";
char *p = "ABCDE";
```

SIRURI DE CARACTERE. OPERARE

*/*Metoda 1*/*

În situația în care constanta șir depășește lungimea unei linii de terminal, fie în cadrul unei inițializări, fie în cazul apelului unui `printf()`, continuarea ei pe linia următoare se face adăugând un `\` la sfârșitul liniei curente.

*/*Metoda 2*/*

O altă variantă constă în divizarea constantei șir inițială (lungă) în două subsiruri alăturate. În C, două constante șir alăturate (separate doar prin spații, tab, enter) sunt concatenate.

Exemplu

```
/*Constanta originala*/
printf("Acesta este un exemplu cu o constanta sir foarte mare ");

/*Metoda 1*/
printf("Acesta este un exemplu cu o \
constanta sir foarte mare ");

/*Metoda 2*/
printf("Acesta este un exemplu cu o "
      "constanta sir foarte mare "
      );
```

CITIRE ȘI AFIȘARE

Pentru a citi un șir de la tastatură se poate folosi funcția `scanf()`.

Aceasta va citi caractere până când este apăsată tasta ENTER, dar șirul va reține caractere până la întâlnirea caracterului spațiu ' ' sau cel de linie nouă (furnizat de apăsarea tastei ENTER). Tasta ENTER este înlocuită de compilator cu caracterul `'\0'` (null) care termină șirul.

Pentru afișare se poate folosi funcția `printf()` și specificatorul `"%s"`.

Exemplu:

```
char str[25];
printf("Introduceti un sir: ");
scanf("%s", str);
printf("Ati introdus sirul <%s>.", str);
```

Pentru a citi un șir de la tastatură se poate folosi și funcția `gets()`.

Aceasta va citi caractere până când este apăsată tasta ENTER, iar șirul va reține caractere până la întâlnirea caracterului de linie nouă (furnizat de apăsarea tastei ENTER). Tasta ENTER este înlocuită de compilator cu caracterul `'\0'` (null) care termină șirul.

Pentru afișare se poate folosi funcția `puts()`. Aceasta va introduce automat un caracter de tip "linie nouă" (`'\n'`) după șirul afișat.

Exemplu:

```
char str[25];
printf("Introduceti un sir: ");
gets(str);
```

```
puts("Ati introdus: ");  
puts(str);
```

CURĂȚAREA BUFFERULUI DE INTRARE:

Dacă se încearcă citirea consecutivă a unor stringuri, este posibil ca anumite caractere netrate să rămână în bufferul de intrare al sistemului (de exemplu taste ENTER sau chiar șiruri de caractere ce încep cu caracterul spațiu ce nu au fost preluate de funcția `scanf()`). Utilizarea unei funcții de preluare a caracterelor de la consola poate duce la neplăcuta surpriză de citire a unor caractere nedorite. Pentru evitarea acestei situații se poate folosi funcția de curățare a bufferelor sistemului, specificând ca și parametru intrarea standard (`stdin`):

```
fflush(stdin);
```

TABLOURI DE ȘIRURI:

Tablourile de șiruri sunt des folosite în C. O tabelă bidimensională de șiruri se poate crea ca oricare alt tablou bidimensional. Totuși modul în care trebuie gândit un tablou de șiruri este puțin diferit. Fie următoarea declarație :

```
char names[10][25];
```

Această declarație specifică o tabelă care conține 10 șiruri, fiecare având lungimea de până la 25 de caractere. Pentru a accesa un șir din tabelă, trebuie specificat numai primul index.

Exemplu:

```
char names[10][25];  
int i;  
  
for (i = 0; i < 3; i++) {  
    fflush(stdin);  
    printf(„Introduceți numele [%d]: ”, i);  
    gets(names[i]);  
}  
  
for (i = 0; i < 3; i++) {  
    printf(„\nnumele %d: <%s>”, i, names[i]);  
}
```

PROBLEME REZOLVATE

Scrieți programul care citește un șir de caractere și va număra câte spații s-au introdus.

```
#include <stdio.h>
#define N 80

int main ( void ) //In Linux este obligatoriu returnarea unei valori
{
    char s[ N ];
    int i, nr_space = 0;
    puts( "\nIntroduceti sirul de caractere dorit" );
    gets( s );
    for( i = 0; s[i]; i++ )
        if( s[ i ] == ' ' )
            nr_space++;
    printf( "\n S-au introdus %d blank-uri", nr_space );
    return 0;
}
```

PROBLEME PROPUSE SPRE REZOLVARE

1. Scrieți un program care să citească un șir de caractere și care să afișeze șirul astfel încât după fiecare caracter să urmeze câte o virgulă.

2. Scrieți un program care să citească un șir de caractere și apoi să afișeze la rând nou câte un subșir care se obține din precedentul prin îndepărtarea primei litere.

3. Se vor citi:

- n = numărul de șiruri de caractere ce vor fi citite;
- respectiv, cele n șiruri de caractere.

Se va calcula suma lungimilor acestor șiruri. Nu se va folosi memorie decât pentru un singur șir.

4. Scrieți un program care să determine dacă un cuvânt dat se găsește într-o propoziție dată. Cuvântul și propoziția vor fi citite de la tastatură.

5. Să se determine dacă un șir de caractere reprezentând o propoziție este negativă sau afirmativă. Veți considera că o propoziție este negativă dacă în conținutul ei apare cuvântul "nu" sau șirul "n-" de un număr impar de ori.

6. Să se citească mai multe șiruri de caractere terminate prin <CR> și să se afișeze șirul de lungime maximă.