Introducere în Programarea calculatoarelor
Laborator 5

Obiectiv: examinarea posibilităţilor lucrului cu şiruri de caractere în limbajul C**.**

Activităţi

- Prezentarea generalităţilor despre lucrul cu şiruri de caractere în limbajul C;
- Lucrul cu principalele funcţii din biblioteca standard C pentru şiruri de caractere;

ŞIRURI DE CARACTERE

Generalități despre lucrul cu șiruri de caractere în limbajul C

Limbajul C nu dispune de un tip de date nativ pentru reprezentarea şirurilor de caractere de lungime variabilă. În acest scop se utilizează structuri de date de tip tablou de caractere.

Întrucât şirurile de caractere prelucrate în programe au în general lungime variabilă, s-a stabilit o convenţie prin care ultimul caracter utilizat al unui şir este urmat de un caracter cu valoarea zero ('\0'), numit terminator de şir.

```
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 şir de caractere de lungime variabilă, dar de maxim 9 de caractere, întrucât ultimul element este rezervat pentru terminatorul de şir.

Dacă şirul de mai sus conţine valoarea "TEST", conţinutul memoriei rezervate tabloului este următorul:

Index	0	1	2	3	4	5	6	7	8	9
Continut	T	E	S	T	**\0**	-	-	-	-	-

Valoarea elementelor tabloului corespunzătoare indicilor de la 5 la 9 nu este cunoscută. Evident, aceste tablouri care memorează şiruri de caractere se pot aloca şi dinamic:

```
char *sir = (char*)malloc (10 * sizeof (char));
/* ...operatii cu variabila sir... */ free (sir);
```

Dacă se doreşte doar accesarea şi prelucrarea elementelor unui şir de caractere care a fost alocat anterior, static (declaraţie de tablou) sau dinamic, se poate utiliza şi o variabilă de tip pointer către caracter:

```
        char sir [10]; char *ptrsir;
        ptrsir = sir;
/*Ambele variabile indica spre acelasi sir de caractere*/
```

Orice şir de caractere care se prelucrează în program trebuie să dispună însă de o declaraţie de alocare de memorie (static sau dinamic).

Compilatoarele de C permit iniţializarea tablourilor de caractere în momentul declarării acestora cu un şir de caractere:

```
char sir [10] = "Un sir";
```

În urma acestei declaraţii, variabila *sir* va avea următorul conţinut:

Index	0	1	2	3	4	5	6	7	8	9
Continut	U	n		S	i	r	**\0**	-	-	-

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

Întrucat nu există tip de date nativ şir de caractere, limbajul nu dispune nici de operatori pentru prelucrarea şirurilor. Există însă o serie de funcţii care asistă programatorul în lucrul cu şirurile de caractere. Aceste funcţii necesită includerea fişierului header **string.h.** Toate funcţiile din această bibliotecă presupun că variabilele de tip şir de caractere asupra cărora operează respectă convenţia de a avea un terminator (caracter cu codul 0).

Citirea și afişarea unui sir de caractere de la tastatură

Pentru citirea unui şir de caractere se poate utiliza funcţia *scanf* cu specificatorul de format %s. Pentru afişarea unui şir de caractere se poate utiliza funcţia printf, cu acelaşi specificator de format:

```
char sir [80];
printf ("Introduceti un sir: ");
scanf ("%s", sir);
printf ("Ati tastat: %s \n", sir);
```

Întrucât variabila şir este declarată ca şi tablou, numele acesteia reprezintă un pointer către primul element, de aceea nu se mai foloseşte operatorul adresă (&) în apelul scanf.

Introducere în Programarea calculatoarelor
Laborator 5

Utilizarea funcţiei scanf pentru citirea şirurilor de caractere are un dezavantaj major: nu se pot citi şiruri care conţin spaţii sau tab. De aceea, se recomandă utilizarea funcţiei *gets*:

```
char sir [80];
 printf ("Introduceti un sir: "); gets (sir);

printf ("Ati tastat: %s \n", sir);
```

Lungimea unui șir de caractere

Pentru a determina lungimea unui şir de caractere se foloseşte funcţia:

```
int strlen (char *sir)
```

Funcţia returnează lungimea şirului de caractere primit ca şi parametru.

Exemplu:

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

Copierea conţinutului unui șir de caractere

Pentru a copia conţinutul unei variabile sau constante de tip şir de caractere într-o variabilă tot de tip şir de caractere trebuie utilizată funcţia:

```
char *strcpy (char *destinatie, char *sursa)
```

Exemplu:

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

Este sarcina programatorului să se asigure că destinaţia are suficient spaţiu alocat pentru a memora toate caracterele din variabila sursă (inclusiv terminatorul de şir).

Introducere în Programarea calculatoarelor
Laborator 5

Compararea alfabetică a două șiruri de caractere

Pentru compararea a două şiruri de caractere nu se pot aplica operatorii relationali între cele doua şiruri (<, >, <=, >=, ==, !=), întrucât aceştia au ca efect compararea adreselor în memorie ale celor două şiruri. De aceea trebuie utilizată această funcţie:

```
int strcmp (char *s1, char *s2)
```

Rezultatul funcţiei este:

- negativ dacă şirul s1 este mai mic decât s2 din punct de vedere al conţinutului;
- zero dacă s1 şi s2 au acelaşi conţinut;
- mai mare ca zero, dacă şirul s1 este mai mare decât şirul s2 din punct de vedere al conţinutului;

Exemplu:

```
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);
```

Căutarea unui subșir într-un șir

Pentru căutarea unui subsir într-un şir se utilizează functia *strstr*:

```
char *strstr (char *sir, char *subsir)
```

Funcţia returnează un pointer la prima apariţie a conţinutului variabilei *subsir* în conţinutul variabilei *sir* sau valoarea NULL dacă subsirul nu apare în şirul căutat.

Exemplu:

```
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 șiruri de caractere

Pentru concatenarea a două şiruri de caractere se foloseşte funcţia:

```
char *strcat (char *dest, char *sursa)
```

Din nou, este responsabilitatea programatorului să se asigure că destinaţia are suficient spaţiu alocat pentru a memora toate caracterele din variabila sursă (inclusiv terminatorul de şir).

Exemplu:

```
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);
```

Introducere în Programarea calculatoarelor
Laborator 5

Probleme rezolvate

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

```
#include <stdio.h>
#include <string.h>
#include <ctype.h> /* pentru toupper */

int main (void)
{
   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'))
          printf ("Nume de fata \n");
         else
           printf ("Nume de baiat \n");
return 0;
}
```

Exemplu: următorul program citeşte de la tastatură un şir de caractere şi afişează şirul în ordine inversă:

```
#include <stdio.h>
#include <string.h>
int main (void)
{
  char sir [80];
  int k, n;
  printf ("Introduceti sirul: ");
  gets (sir);
  n = strlen (sir);
  for (k = n - 1; k >= 0; k--)
      printf ("%c", sir [k]);
printf ("\n");
return 0;
}
```

Introducere în Programarea calculatoarelor
Laborator 5

Exemplu: următorul program citeşte un şir de caractere reprezentând un număr în baza 2 şi afişează valoarea acestuia în baza 10:

```
#include <stdio.h>
#include <string.h>
int main (void)
{
  char sir [80];
  int k, n, val, p, cifra, valid;
        do {
          do
            {
              printf ("Introduceti sirul: ");
              gets (sir);
              n = strlen (sir);
             }while (n == 0);

          valid = 1;

          for (k = 0; k < n; k++)
               if ((sir [k] != '0') && (sir [k] != '1'))
                          valid = 0;
               if (!valid)
                  printf ("Sirul introdus nu este numar in  baza 2 ! \n");
        }while (!valid);
        p = 1;      /* puterea lui 2 */
        val = 0;    /* valoarea numarului */

         for (k = n - 1; k >= 0; k--)
        {
           cifra = sir [k] – '0';
           val = val + cifra * p;
           p = p * 2;
        }
        printf ("Numarul %s in baza 10 este: %d \n",sir,val);
        return 0;
}
```

Probleme propuse

1. Să se implementeze funcţiile strlen, strcpy, strcmp, strstr, strcat.

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

Introducere în Programarea calculatoarelor
Laborator 5

3. Să se scrie un program C care realizează următoarele:
   a) Şterge dintr-un şir de caractere un subşir specificat prin poziţie şi lungime.
   b) Inserează într-un şir începând cu o poziţie dată un al şir.
   c) Citeşte doua cuvinte şi înlocuieşte într-un text introdus de la tastatură toate apariţiile primului cuvânt prin cel de-al doilea.

4. Să se citească un text de la tastatură care se încheie cu '.'. Să se afişeze cel mai lung cuvânt din text (dacă sunt mai multe cu aceeaşi lungime maximă se va afişa doar unul dintre ele).