Şiruri de caractere

Formă generală. Declarare

1. **Funcţii specifice şirurilor de caractere**
2. **Probleme cu şiruri de caractere**

Formă generală. Declarare

- Datele care se reprezintă sub formă se şiruri de caractere au o largă aplicabilitate în programarea calculatoarelor, indiferent de limbajul folosit.
- Astfel şi în limbajul C/C++ se pot memora şi prelucra informaţii de tip şir de caractere.
- Cu toate că *limbajul C/C++ nu conţine un tip de date special pentru şiruri de caractere* aşa cum are limbajul Pascal, ***se pot utiliza tablouri unidimensionale de caractere***.

Formă generală. Declarare

Declararea unui tablou de caractere se face astfel:

```
char nume_tablou[dimensiune_maximă];
```

Exemple:

```
char sir[20];      // tablou  de 20 de caractere
char t[10];        // tablou  de 10 caractere
```

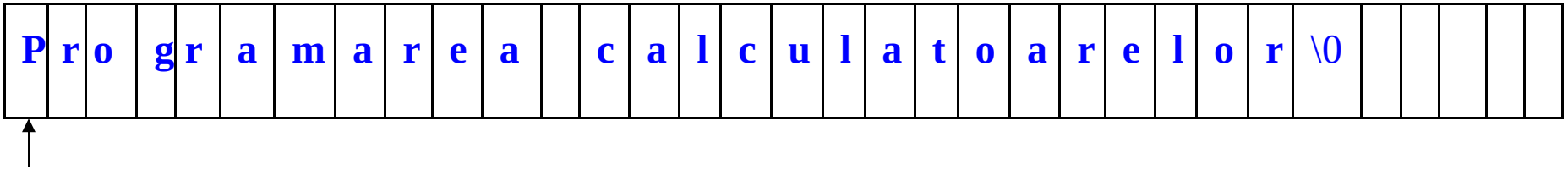
- Pentru a specifica sfârşitul şirului de caractere, după ultimul caracter se adaugă un octet cu valoarea 0 (caracterul **'\0'**).

Formă generală. Declarare

- Dimensiunea declarată pentru un tablou de şiruri de caractere trebuie să fie cu o unitate mai mare pentru ca pe ultima poziţie să se poată pune şi valoarea **'\0' = terminatorul de şir**.

  Reprezentarea internă a unui şir de caractere

char sir[33]="Programarea calculatoarelor";



sir[0] **sir[33]="Programarea calculatoarelor"**

Ultimi octeţi sunt nefolosiţi.

Formă generală. Declarare

- În limbajul C/C++, *un şir de caractere este un tablou unidimensional cu elemente de tip caracter şi care se termină cu* ***NULL***.
- Totuşi compilatorul C/C++ nu adaugă automat terminatorul **NULL**, decât în cazul folosirii funcţiilor predefinite *fgets()* şi *gets() (functii de citire din fisiere sau de la tastatura),* iar în celelalte cazuri *este necesar ca programatorul să adauge terminatorul de şir* atunci când doreşte acest lucru, pentru a lucra cu şirurile de caractere.

Formă generală. Declarare

Exemplu:

Următorul program declară un şir de caractere de 256 de elemente şi atribuie primelor 26 de locaţii libere literele mari ale alfabetului:

```
#include<iostream.h>
int main()
{
      char sir[256];
      int i;
      for(i=0;i<26;i++)
            sir[i] = 'A' + i;
      sir[i] = '\0';
     cout<<"Sirul de caractere contine: "<<sir;
}
```

Funcţii specifice şirurilor de caractere
Şiruri de caractere

1. Formă generală. Declarare

2. Funcţii specifice şirurilor de caractere

3. Probleme cu şiruri de caractere

În biblioteca limbajului C++ există câteva funcţii specifice şirurilor de caractere:

Funcţii specifice şirurilor de caractere

- ✓în fişierul standard de intrare / ieşire – ***stdio.h***, avem funcţiile *gets()* şi *puts().*
- ✓în fişierul ***string.h***, avem mai multe funcţii pe care le voi prezenta în continuare.

Pentru *citirea sirurilor de caractere care contin spatii* se poate folosi metoda **getline** a funcției **cin**:

```
cin.getline(variabila_sir, dimensiune_maxima);
```

Funcţii specifice şirurilor de caractere

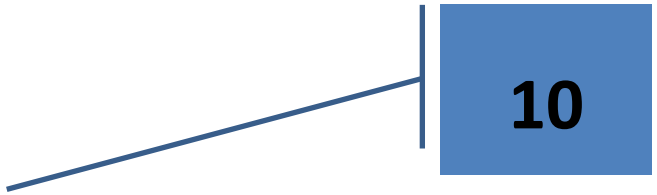
Exemplu:

char variabila_sir[120]; cin.getline(variabila_sir, 120); Funcţiile pentru operaţii cu şiruri ce se găsesc în header-ul **<string.h>.**

strlen (nume_şir**)**

Returnează un număr întreg ce reprezintă lungimea unui şir de caractere, fără a număra terminatorul de şir.

Funcţii specifice şirurilor de caractere



Exemplu:

```
cout <<strlen("sir corect");
```

Funcţii specifice şirurilor de caractere

strcmp (şir_1, şir_2**)**

Funcţia compară cele două şiruri date ca argument şi returnează o valoare întreagă egală cu diferenţa dintre codurile ASCII ale primelor caractere care nu coincid.

0

Exemplu:

```
cout <<strcmp("carte", "carte");
```

Funcţii specifice şirurilor de caractere

strcpy (şir_destinaţie, şir_sursă**)**

Funcţia copiază şirul sursă în şirul destinaţie.

Nota:

Pentru a fi posibilă copierea, lungimea şirului destinaţie trebuie să fie mai mare sau egală cu cea a şirului sursă, altfel pot aparea erori grave.

Exemplu:

"carte_informatica"

Funcţii specifice şirurilor de caractere

cout <<strcpy(sir, "carte_informatica");

strcat (şir_destinaţie, şir_sursă**)**

Funcţia concatenează cele două şiruri: şirul sursă este adăugat la sfârşitul şirului destinaţie.

Tabloul care conţine şirul destinaţie trebuie să aibă suficiente elemente.

Exemplu:

```
char sir[]="carte";
    cout <<strcat(sir, "informatica");
```

"carteinformatica"

Funcţii specifice şirurilor de caractere

strchr(sir, caracter**)**

Returneaza o valoare pozitiva daca un caracter apare intr-un sir, 0 in caz contrar.

Exemplu:

```
if (strchr(sir, c)) cout << "L-am gasit!"
```

Funcţii specifice şirurilor de caractere

strstr(sir1, sir2**)**

Returneaza o valoare pozitiva daca un sir apare intr-un alt sir, 0 in caz contrar.

Exemplu:

```
if (strstr(sir1,sir2)) cout <<"Am gasit subsirul!"
```

Exemplu 1:

```
#include <iostream.h>
#include <string.h> int
main()
{
  char sir1[] = "abcd", sir2[] = "abcde";
cout<<strcmp(sir1, sir2)<<"\n";
        // afişare: -101
        // ’a’ = 97,…, ’d’ = 100, ’e’ = 101
        // ’0’ - ’e’ =  -101
```

Funcţii specifice şirurilor de caractere

```
cout<<strcmp(sir2, sir1)<<"\n";
// afişare: 101

cout<<strcmp(sir1, "")<<"   ";
// compararea variabilei sir1 cu constanta şir vid
char str1[20]="hello";
charstr2[20]="goodbye";
char str3[20]="";
```

Funcţii specifice şirurilor de caractere

```
int diferenta, lungime;
cout<<"str1="<<str1<<"        str2="<<str2<<"\n";
diferenta = strcmp(str1, str2);
if (diferenta == 0)
   cout<<"Siruri echivalente! "<<"\n";
       else
   if (diferenta > 0)
    cout<<str1<<" mai mare decât "<<str2<<"\n";
else
    cout<<str1<<" mai mic decât "<<str2<<"\n";
```

Funcţii specifice şirurilor de caractere

```
 cout<<"str1="<<str1<<"\n";
 cout<<"str3="<<str3<<"\n";
strcpy (str3, str1);
cout<<"str1="<<str1<<"\n";
cout<<"str3="<<str3<<"\n";
strcat (str3, str1);
cout<<"str1="<<str1<<"\n";
cout<<"str3="<<str3<<"\n";
}
```

Funcţii specifice şirurilor de caractere

Exemplu 2:

Verificati daca un nume apare intr-un sir.

```
#include <iostream.h>
#include <string.h>
int main()
{
   char nume[10],sir[100];
```

```
 cout<<"Introduceti numele: ";
    cin.getline(nume,10);
 cout<<"Introduceti sirul: ";
    cin.getline(sir,100);
 if(strstr(sir,nume))
   cout<<nume<<" apare in sirul "<<sir<<"\n";
        else
    cout<<nume<<" NU apare in sirul "<<sir<<"\n";
}
```

În fişierul **<stdlib.h>** există câteva funcţii care prelucrează şirurile de caractere.

Acestea sunt *funcţiile de conversie dintr-un număr într-un şir de caractere şi invers*.

Funcţii specifice şirurilor de caractere

În programul următor vor fi exemplicate funcţiile de conversie şiruri de caractere în numere:

Funcţie	la ce foloseşte
atof	converteşte un şir de caractere într-un număr real simplă precizie
atoi	converteşte un şir de caractere într-un număr întreg
atol	converteşte un şir de caractere într-un număr întreg de tip long
strtod	converteşte un şir de caractere într-un număr real dublă precizie
strtol	converteşte un şir de caractere într-un număr de tip long

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

Funcţii specifice şirurilor de caractere

```
#include<stdlib.h>
int main()
{
    int numar_int;
float numar_real;
long numar;
numar_int = atoi("6789");
    numar_real = atof("12.345");
    numar = atol "1234567890L")
  cout<<numar_int<<" "<<numar_real<<" "<<numar; }
```

Funcţii specifice şirurilor de caractere

În programul următor vor fi exemplicate funcţiile de conversie numere în şiruri de caractere:

Funcţie	la ce foloseşte
itoa	converteşte un număr întreg într-un şir de caractere
ftoa	converteşte un număr real simplă precizie într-un şir de caractere
ultoa	converteşte un număr de tip long unsigned într-un şir de caractere

Funcţii specifice şirurilor de caractere

```
#include<iostream.h
> #include<stdlib.h>
int main()
{
   int numar_int = 6789;
   long numar = 1234567890L;
   char sir[25];
   itoa(numar_int, sir, 10);
   cout<<" numar = "<< numar_int<<" sir = "<<sir;
   ltoa(numar, sir, 10);   cout<<" numar = "<<
numar<<" sir = "<<sir;
```

Funcţii specifice şirurilor de caractere

}

1. Funcţii specifice şirurilor de caractere
2. Probleme cu şiruri de caractere

Probleme cu şiruri de caractere

Problema 1:

Enunţ:

Să se afişeze numărul de vocale dintr-un text scris cu litere mici, memorat intr-o variabilă de tip șir de caractere.

Exemplu:

Date de intrare: Programarea calculatoarelor

Date de ieșire: Exista 12 vocale in text

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

Probleme cu şiruri de caractere

```
#include <string.h>

int main()
{
 char text[100];
char vocale[] = "aeiou";
 int        contor        =        0;
cout<<"Introduceti   text:   ";
cin.getline(text, 100);
     for (int i=0; i<strlen(text); i++)
          for (int j=0; j<strlen(vocale); j++)
```

```
        if (text[i] == vocale[j])
contor++;
   cout<<"Exista "<<contor<<" vocale in text. ";
}
```

Problema 2:

Enunţ:

Să se afişeze cu litere mari un text dat, de maxim 255 caractere.

Probleme cu şiruri de caractere

Exemplu:

Date de intrare text: liMBajUl c++

Date de ieşire LIMBAJUL C++

```
#include <iostream.h>

int main()
{
 char sir[255];  cout<<"Introduceti
textul: ";
 cin.getline(sir, 255);
```

Probleme cu şiruri de caractere

```
 for (int i=0; sir[i]!=0; i++)
  if (sir[i] >= 'a' && sir[i] <= 'z')
      sir[i] = sir[i] - 'a' + 'A';
cout<<sir;
}
```

Problema 3:

Enunţ:

Se citește un șir de caractere. Să se afișeze litera cea mai des întâlnită.

Exemplu:

Probleme cu şiruri de caractere

Date de intrare text: carte de informatica si programare

Date de ieşire: a apare de 5 ori

```
#include <string.h>

char sir[1000], carac_max, c;  int i, frecv[256],
max=0;  // frecv - vector de frecvente
  cout<<"Dati sirul de caractere: ";
cin.getline(sir,1000);
    for (i=0; i<256; i++) frecv[i] = 0;
```

Probleme cu şiruri de caractere

```
for (i=0;i<strlen(sir);i++)
        {
        c=sir[i];
        frecv[c]=frecv[c]+1;
        if (frecv[c]>max){
                  max=frecv[c];
                 carac_max=c;
                 }
    }
cout<<carac_max<<" apare de "<<max<<" ori";
```

1. Să se afișeze toate prefixele și sufixele unui cuvânt citit de la tastatură.

Exemplu: **Date de intrare: informatica**

Date de ieşire:

i
in
inf
info
infor
inform
informa
informat
informati
informatic
informatica

a
ca
ica
tica
atica
matica
rmatica
ormatica
formatica
nformatica
informatica

2. Se citește de la tastatură un caracter c și un text de maxim 100 de carectere. Afișați de câte ori apare caracterul c în cadrul textului. Literele mari se vor considera diferite de literele mici.

Exemplu:

Date de intrare:

Acesta este un simplu exemplu

Caracterul e

Date de ieşire:

Litera ‘e’ apare de 5 ori

3. Se citește de la tastatură un șir de maxim 100 de carectere format numai din litere și cifre. Afișați numărul literelor mari, numărul literelor mici și numărul caracterelor de tip cifră din textul dat.

Exemplu:

Date de intrare:

Brâncuși s-a nascut pe 18 februarie 1876 în Gorj

Hobita

Date de ieşire:

Textul are 36 litere mici, 3 litere mari si 6 cifre.