

Fundamentele Programarii – Laborator 8

Siruri de caractere

Obiective

La finalul acestui laborator, studentul va fi capabil sa:

- inteleaga reprezentarea sirurilor de caractere in C++;
- utilizeze corect siruri de tip `char[]`;
- foloseasca tipul `std::string`;
- aplice operatii uzuale asupra sirurilor de caractere.

Notiuni teoretice

Siruri de caractere de tip `char[]`

Un sir de caractere este un tablou de caractere terminat cu caracterul special `\0`.

```
1 char s[20] = "programare";
```

Caracterul `\0` marcheaza sfarsitul sirului.

Citirea sirurilor de tip `char[]`

```
1 char s[50];  
2 cin >> s;
```

Aceasta metoda citeste pana la primul spatiu.

Funcții din biblioteca `cstring`

- `strlen(s)` – lungimea sirului;
- `strcpy(a,b)` – copierea unui sir;
- `strcmp(a,b)` – compararea sirurilor;
- `strcat(a,b)` – concatenarea sirurilor.

Tipul `std::string`

Declarare si initializare

```
1 #include <string>  
2 string s = "informatica";
```

Citirea cu spatii

```
1      getline(cin, s);
```

Operatii uzuale

- `s.length()`
- `s + t`
- `s == t`
- `s[i]`

Exemple

Lungimea unui sir

```
1      string s;  
2      getline(cin, s);  
3      cout << s.length();
```

Numararea vocalelor

```
1      int cnt = 0;  
2      for (int i = 0; i < s.length(); i++) {  
3          if (s[i]=='a' || s[i]=='e' || s[i]=='i' || s[i]=='  
4              'o' || s[i]=='u')  
5              cnt++;  
6      }  
7      cout << cnt;
```

Exercitii propuse

Exercitii de baza

1. Cititi un sir si afisati lungimea lui.
2. Afisati sirul inversat.

Exercitii aplicative

1. Numarati cate vocale contine un sir.
2. Verificati daca un sir este palindrom.

Exercitii bonus

1. Eliminati toate spatiile dintr-un sir.
2. Determinati cuvantul cel mai lung dintr-o propozitie.

Observatii

Tipul `std::string` este mai sigur si mai usor de utilizat decat sirurile clasice de tip `char[]`.

Laboratorul urmator: Functii