

Programarea calculatoarelor

10 ^{C++} Siruri de caractere

Adrian Runceanu

www.runceanu.ro/adrian

Curs 10

Capitolul 8. Șiruri de caractere

8.1. Formă generală. Declaraire

8.2. Funcții specifice șirurilor de caractere

8.3. Probleme cu șiruri de caractere

8.1. Formă generală. Declaraire

- Datele care se reprezintă sub formă de șiruri de caractere au o largă aplicabilitate în programarea calculatoarelor, indiferent de limbajul folosit.
- Astfel și în limbajul C/C++ se pot memora și prelucra informații de tip șir de caractere.
- Cu toate că *limbajul C/C++ nu conține un tip de date special pentru șiruri de caractere* așa cum are limbajul Pascal, *se pot utiliza tablouri unidimensionale de caractere.*

8.1. Formă generală. Declaraire

Declaraarea 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
```

8.1. Formă generală. Declaraire

- Pentru a specifica sfârșitul șirului de caractere, după ultimul caracter se adaugă **un octet cu valoarea 0** (caracterul **'\0'**).
- **Dimensiunea declarată** pentru un tablou de șiruri de caractere trebuie să fie **cu o unitate mai mare** pentru ca pe ultima poziție să se poată pune și valoarea **'\0' = terminatorul de șir**.

8.1. Formă generală. Declaraire

Reprezentarea internă a unui șir de caractere

```
char sir[33]="Programarea calculatoarelor";
```

P	r	o	g	r	a	m	a	r	e	a			c	a	l	c	u	l	a	t	o	a	r	e	l	o	r	\0						
---	---	---	---	---	---	---	---	---	---	---	--	--	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	----	--	--	--	--	--	--



sir[0]

sir[33]="Programarea calculatoarelor"

Ultimi octeți sunt nefolosiți.

8.1. Formă generală. Declaraire

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

8.1. Formă generală. Declaraire

Exemplu:

Următorul program declară un șir de caractere de 256 de elemente și atribuie primelor 26 de locații 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;
}
```



Codurile ASCII:
['a', 'z'] = [97, 122]
['A', 'Z'] = [65, 90]

După compilarea și execuția programului, pe ecran vor apărea următoarele valori:

Capitolul 8. Șiruri de caractere

8.1. Formă generală. Declarație

8.2. Funcții specifice șirurilor de caractere

8.3. Probleme cu șiruri de caractere

8.2. Funcții specifice șirurilor de caractere

În biblioteca limbajului C++ există câteva funcții specifice șirurilor de caractere:

- ✓ în fișierul standard de intrare / ieșire – ***stdio.h***, avem funcțiile ***gets()*** și ***puts()***.
- ✓ în fișierul ***string.h***, avem mai multe funcții specifice prelucrării șirurilor, funcții pe care le voi prezenta în continuare.

8.2. Funcții specifice șirurilor de caractere

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

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

Exemplu:

```
char variabila_sir[120];  
cin.getline(variabila_sir, 120);
```

8.2. Funcții specifice șirurilor de caractere

Funcțiile pentru operații cu șiruri ce se găsesc în header-ul **<string.h>**.

strlen (nume_șir)

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

Exemplu:

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



10

8.2. Funcții specifice șirurilor de caractere

strcmp (șir_1, șir_2)

Funcția compară cele două șiruri date ca argument și returnează o valoare întreagă egală cu diferența dintre codurile ASCII ale primelor caractere care nu coincid.

Exemplu:

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



0

8.2. Funcții specifice șirurilor de caractere

strcpy (șir_destinație, șir_sursă)

Funcția copiază șirul sursă în șirul destinație.

Nota:

Pentru a fi posibilă copierea, lungimea șirului destinație trebuie să fie mai mare sau egală cu cea a șirului sursă, altfel pot apărea erori grave.

Exemplu:

```
char sir[20];
```

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

"carte_informatica"

8.2. Funcții specifice șirurilor de caractere

strcat (șir_destinație, șir_sursă)

Funcția concatenează cele două șiruri: șirul sursă este adăugat la sfârșitul șirului destinație.

Tabloul care conține șirul destinație trebuie să aibă suficiente elemente.

Exemplu:

```
char sir[]="carte";
```

```
cout <<strcat(sir, "informatica");
```

"carteinformatica"

8.2. Funcții specifice șirurilor de caractere

strchr(sir, caracter)

Returnează o valoare pozitivă dacă un caracter apare într-un șir, 0 în caz contrar.

Exemplu:

```
char sir[]="exemplu de sir", c='i';  
if (strchr(sir, c)) cout << "L-am gasit!"
```

8.2. Funcții specifice șirurilor de caractere

strstr(sir1, sir2)

Returnează o valoare pozitivă dacă un șir apare într-un alt șir, 0 în caz contrar.

Exemplu:

```
char sir1="primul sir", sir2="sir";  
if (strstr(sir1,sir2)) cout <<"Am gasit subsirul!"
```

8.2. Funcții specifice șirurilor de caractere

char *strdup(const char*s);

Alocă memorie la o altă adresă și copiază în acea memorie șirul s. Intoarce adresa noului șir

Exemplu:

```
char *s="sir initial";  
char *t=strdup(s);  
cout <<"sirul intors "<<t;
```

8.2. Funcții specifice șirurilor de caractere

Exemplu 1:

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

```
int main()
{
```

```
    char sir1[] = "abcd", sir2[] = "abcde";
    cout<<strcmp(sir1, sir2)<<"\n";
    // afișare: -1
```

strcmp(sir1,sir2)=
-1, daca sir1<sir2
0, daca sir1=sir2
1, daca sir1>sir2

8.2. Funcții specifice șirurilor de caractere

```
cout<<strcmp(sir2, sir1)<<"\n";
```

```
// afișare: 1
```

```
cout<<strcmp(sir1, "")<<"  ";
```

```
// compararea variabilei sir1 cu constanta șir vid
```

```
// afișare: 1
```

8.2. Funcții specifice șirurilor de caractere

```
char str1[20]="hello";  
char str2[20]="goodbye";  
char str3[20]="";  
int diferenta, lungime;  
cout<<"str1="<<str1<<" str2="<<str2<<"\n";
```

8.2. Funcții specifice șirurilor de caractere

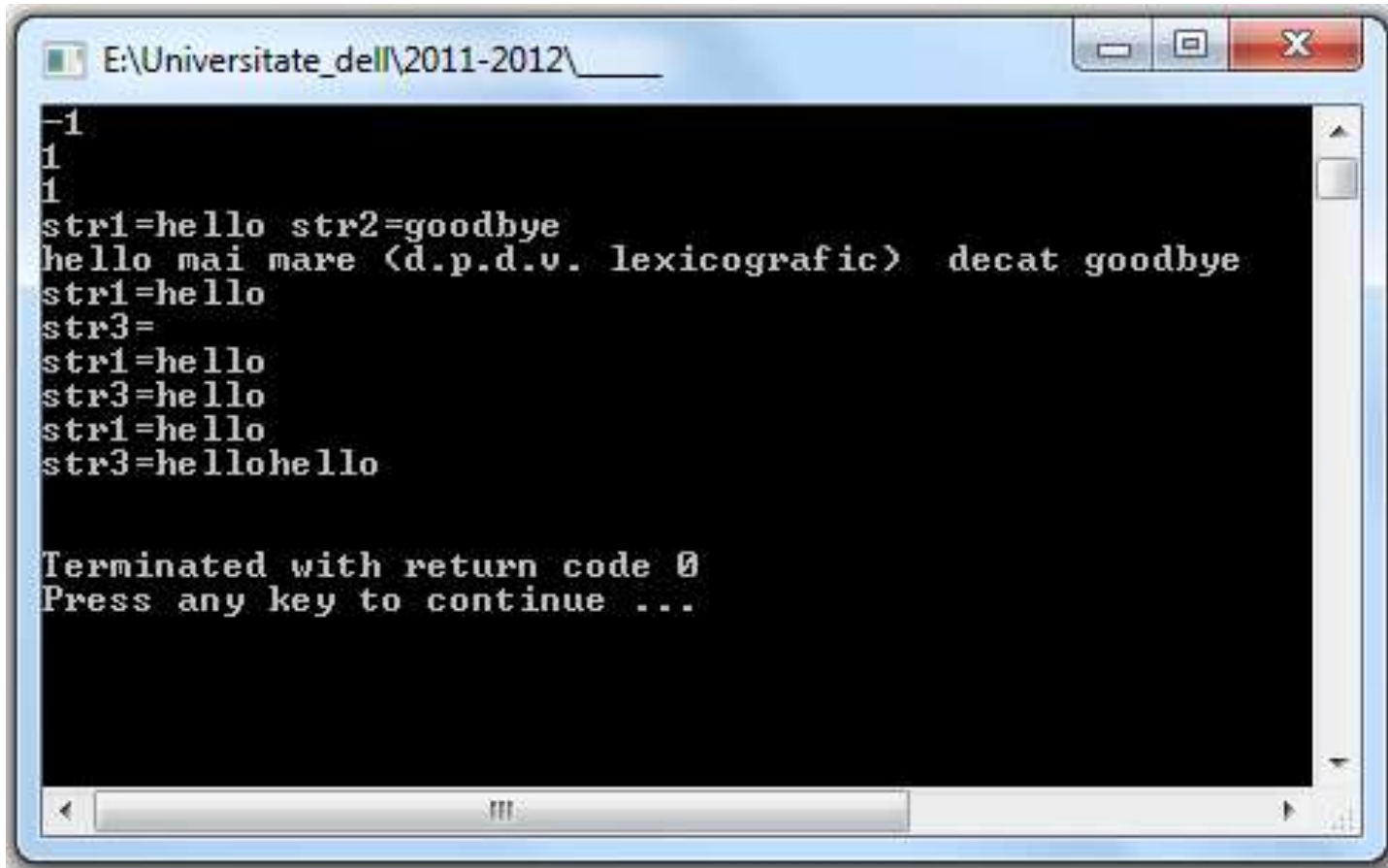
```
diferenta = strcmp(str1, str2);  
if (diferenta == 0)  
    cout<<"Siruri echivalente! "<<"\n";  
else  
    if (diferenta > 0)  
        cout<<str1<<" mai mare (d.p.d.v.  
lexicografic)  decât "<<str2<<"\n";  
    else  
        cout<<str1<<" mai mic (d.p.d.v. lexicografic)  
        decât "<<str2<<"\n";
```

8.2. Funcții specifice șirurilor 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";  
}
```


8.2. Funcții specifice șirurilor de caractere

Executia programului:



```
-1
1
1
str1=hello str2=goodbye
hello mai mare (d.p.d.v. lexicografic) decat goodbye
str1=hello
str3=
str1=hello
str3=hello
str1=hello
str3=hellohello

Terminated with return code 0
Press any key to continue ...
```

8.2. Funcții specifice șirurilor de caractere

Exemplu 2:

Verificati daca un nume apare intr-un sir.

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

```
#include <string.h>
```

```
int main()
```

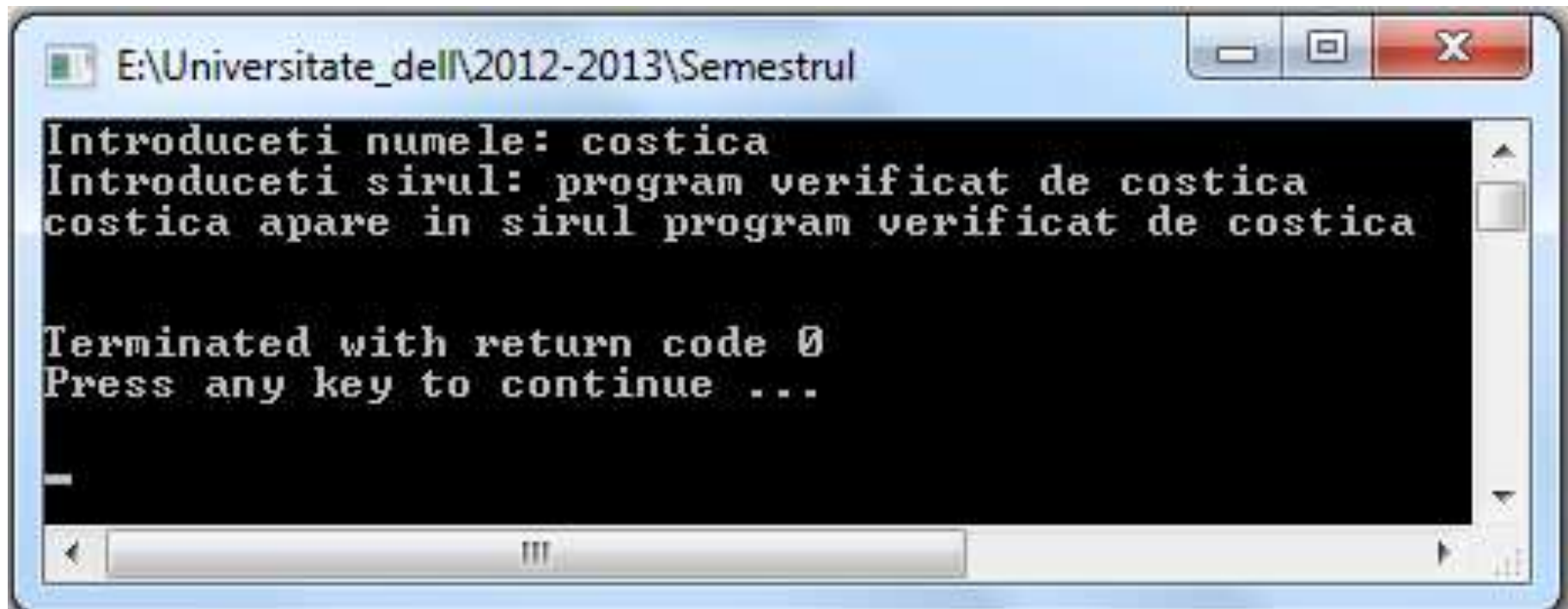
```
{  
    char nume[10],sir[100];  
    cout<<"Introduceti numele: ";  
    cin.getline(nume,10);
```

8.2. Funcții specifice șirurilor de caractere

```
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";
}
```

8.2. Funcții specifice șirurilor de caractere

Executia programului:



```
E:\Universitate_dell\2012-2013\Semestrul

Introduceti numele: costica
Introduceti sirul: program verificat de costica
costica apare in sirul program verificat de costica

Terminated with return code 0
Press any key to continue ...
```

8.2. Funcții specifice șirurilor de caractere

În fișierul **<stdlib.h>** există câteva funcții care prelucrează șirurile de caractere.

Acestea sunt *funcțiile de conversie dintr-un număr într-un șir de caractere și invers*.

8.2. Funcții specifice șirurilor de caractere

În programul următor vor fi exemplificate funcțiile de *conversie din șiruri de caractere în numere*:

Funcție	la ce folosește
<i>atof</i>	convertește un șir de caractere într-un număr real simplă precizie
<i>atoi</i>	convertește un șir de caractere într-un număr întreg
<i>atol</i>	convertește un șir de caractere într-un număr întreg de tip long
<i>strtod</i>	convertește un șir de caractere într-un număr real dublă precizie
<i>strtoul</i>	convertește un șir de caractere într-un număr de tip long

8.2. Funcții specifice șirurilor de caractere

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

```
#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;
```

```
}
```



8.2. Funcții specifice șirurilor de caractere

În programul următor vor fi exemplificate funcțiile de *conversie din numere în șiruri de caractere*:

Funcție

la ce folosește

itoa

convertește un număr întreg într-un șir de caractere

ftoa

convertește un număr real simplă precizie într-un șir de caractere

ultoa

convertește un număr de tip long unsigned într-un șir de caractere

8.2. Funcții specifice șirurilor 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;
}
```

8.2. Funcții specifice șirurilor de caractere

Executia programului:

A screenshot of a Windows command prompt window. The title bar shows the path 'E:\Universitate_dell\2011-2012__'. The command prompt displays the output of a program: 'numar = 6789 sir = 6789 numar = 1234567890 sir = 1234567890'. Below this, it says 'Terminated with return code 0' and 'Press any key to continue ...'. The window has standard Windows controls (minimize, maximize, close) in the top right corner and a scrollbar on the right side.

```
E:\Universitate_dell\2011-2012\__  
numar = 6789 sir = 6789 numar = 1234567890 sir = 1234567890  
Terminated with return code 0  
Press any key to continue ...
```

Capitolul 8. Șiruri de caractere

8.1. Formă generală. Declarație

8.2. Funcții specifice șirurilor de caractere

8.3. Probleme cu șiruri de caractere

8.3. Probleme cu șiruri de caractere

Problema 1:

Enunț:

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

Exemplu:

Date de intrare: Programarea calculatoarelor

Date de ieșire: Exista 12 vocale in text

8.3. Probleme cu șiruri de caractere

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

```
#include <string.h>
```

```
int main()
```

```
{
```

```
    char text[100];
```

```
    char vocale[] = "aeiou";
```

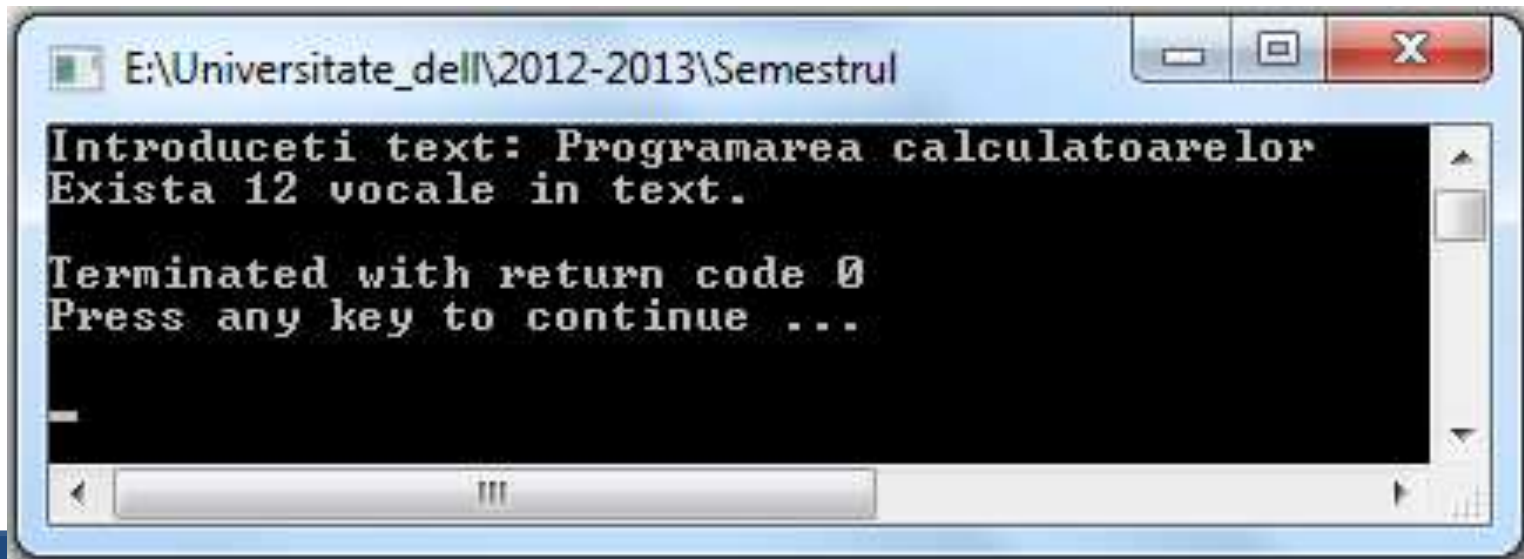
```
    int contor = 0;
```

```
    cout<<"Introduceti text: ";
```

```
    cin.getline(text, 100);
```

8.3. Probleme cu șiruri de caractere

```
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. ";  
}
```



```
E:\Universitate_dell\2012-2013\Semestrul  
Introduceti text: Programarea calculatoarelor  
Exista 12 vocale in text.  
Terminated with return code 0  
Press any key to continue ...
```

8.3. Probleme cu șiruri de caractere

Problema 2:

Enunț:

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

Exemplu:

Date de intrare text: liMBajUl c++

Date de ieșire LIMBAJUL C++

8.3. Probleme cu șiruri de caractere

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

```
int main()
```

```
{
```

```
    char sir[255];
```

```
    cout<<"Introduceti textul: ";
```

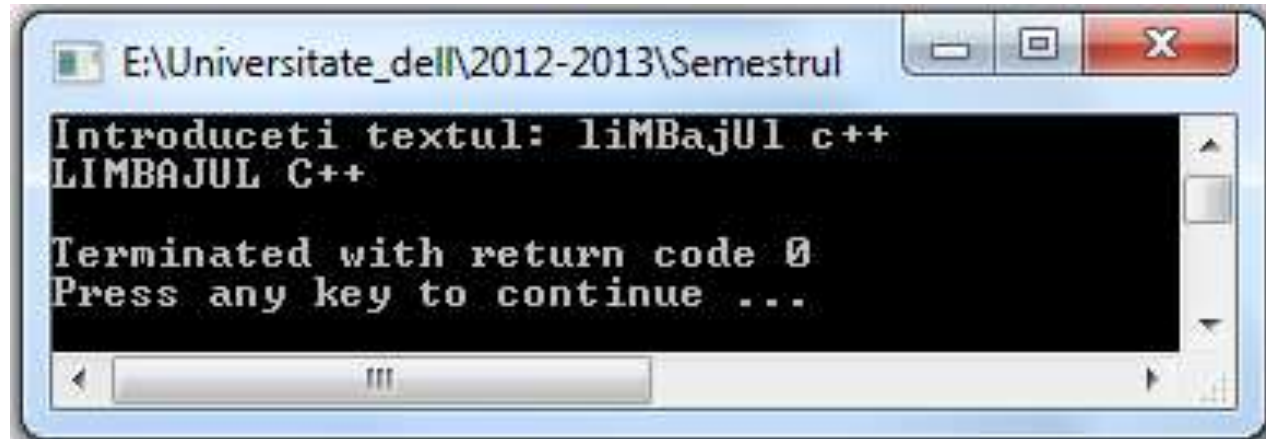
```
    cin.getline(sir, 255);
```

```
    for (int i=0; sir[i]!=0; i++)
```

```
        if (sir[i] >= 'a' && sir[i] <= 'z') sir[i] = sir[i] - 'a' + 'A';
```

```
    cout<<sir;
```

```
}
```



Codurile ASCII:

['a', 'z'] = [97, 122]

['A', 'Z'] = [65, 90]

8.3. Probleme cu șiruri de caractere

Problema 3:

Enunț:

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

Exemplu:

Date de intrare text: **c**arte de inform**a**tica si
program**a**re

Date de ieșire: **a** apare de 5 ori

8.3. Probleme cu șiruri de caractere

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

```
#include <string.h>
```

```
int main()
```

```
{
```

```
    char sir[1000], carac_max, c;
```

```
    int i, freqv[256], max=0;           // freqv - vector de frecvente
```

```
    cout<<"Dati sirul de caractere: ";
```

```
    cin.getline(sir,1000);
```

```
    for (i=0; i<256; i++) freqv[i] = 0;
```

8.3. Probleme cu șiruri de caractere

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

8.3. Probleme cu șiruri de caractere



A screenshot of a Windows command prompt window. The title bar shows the path "E:\Universitate_de\2012-2013\Semestrul". The command prompt displays the text "Dati sirul de caractere: carte de informatica si programare" followed by "a apare de 5 ori" on the next line. Below this, it says "Terminated with return code 0" and "Press any key to continue ...". The window has standard Windows controls (minimize, maximize, close) in the top right corner and a scrollbar on the right side.

```
E:\Universitate_de\2012-2013\Semestrul  
Dati sirul de caractere: carte de informatica si programare  
a apare de 5 ori  
Terminated with return code 0  
Press any key to continue ...
```

Probleme propuse spre rezolvate – șiruri de caractere

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

Exemplu:

Date de intrare: informatica

Date de ieșire:

i
in
inf
info
infor
inform
informa
informat
informati
informatic
informatica

a
ca
ica
tica
atica
matica
rmatica
ormatica
formatica
nformatica
informatica

Probleme propuse spre rezolvate – șiruri de caractere

2. Se citește de la tastatură un caracter c și un text de maxim 100 de caractere. 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 ieșire:

Litera 'e' apare de 5 ori

Probleme propuse spre rezolvate – șiruri de caractere

3. Se citește de la tastatură un șir de maxim 100 de caractere 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:

S-a nascut la Targu Jiu in 1968 pe 21 martie

Date de ieșire: Se gasesc 25 de litere mici, 2 litere mari si 6 cifre

Pentru alte informații teoretice și aplicative legate de acest capitol se recomandă următoarele referințe bibliografice:

1. Adrian Runceanu, Mihaela Runceanu, ***Noțiuni de programare în limbajul C++***, Editura Academica Brâncuși, Târgu-Jiu, 2012 (www.utgjiu.ro/editura)
2. Adrian Runceanu, **Programarea și utilizarea calculatoarelor**, Editura Academica Brâncuși, Târgu-Jiu, 2003 (www.utgjiu.ro/editura)
3. Octavian Dogaru, **C++ - teorie și practică**, volumul I, Editura Mirton, Timișoara, 2004 (www.utgjiu.ro/editura)

Întrebări?