

# Sesiunea 7

---

## Obiective

- Recapitulare probleme manipulare siruri de caractere
- Recapitulare teorie structuri de date

## Exercitii manipulare siruri de caractere

- Sursa exercitii: Primele 10 probleme de la: [https://www.pbinfo.ro/?pagina=probleme-lista&disciplina=0&clasa=10&tag=10&subtag=11&dificultate=1&folosesc\\_consola=-1&eticheta=](https://www.pbinfo.ro/?pagina=probleme-lista&disciplina=0&clasa=10&tag=10&subtag=11&dificultate=1&folosesc_consola=-1&eticheta=)

1. <https://www.pbinfo.ro/probleme/4184/bacnume>

- Date intrare: "David Popovici"
- Date iesire: "Popovici2022"
- Solutie:

```
#include <iostream>
#include <string.h>
using namespace std;

void FNum(char s[], char id[]);

int main(){
    char s[51], id[51];
    cin.getline(s, 51);
    FNum(s, id);
    cout << id;
}

void FNum(char s[], char id[]) {
    char* pch = strtok(s, " ");
    pch = strtok(NULL, " ");
    strcpy(id, pch);
    strcat(id, "2022");
}
```

2. <https://www.pbinfo.ro/probleme/972/pozitii>

- Date intrare: "oasele sunt fragile"
- Date iesire: 4
- Solutie:

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

```
using namespace std;

int este_consoana(char ch);
int este_vocala(char chr);

int main()
{
    char sir[256];
    int contor = 0;
    cout << "Introduceti sirul:";
    cin.getline(sir, 256);
    for (int i = 1; i < strlen(sir) - 1; i++)
    {
        if (este_vocala(sir[i]) && este_consoana(sir[i - 1]) &&
            este_consoana(sir[i + 1]))
        {
            contor++;
        }
    }
    cout << contor;

    return 0;
}

int este_vocala(char chr)
{
    if (strchr("aeiou", chr) != NULL)
    {
        return 1;
    }
    else
    {
        return 0;
    }
}

int este_consoana(char ch) {
    if (ch > 'a' && ch <= 'z' && !este_vocala(ch)) {
        return 1;
    }
    else {
        return 0;
    }
}
```

3. <https://www.pbinfo.ro/probleme/11/vocale>

- Date intrare: "romancier"
- Date iesire: "rOmAncIEr"
- Solutie:

```

#include <iostream>
#include <cstring>

using namespace std;

int isVowel(char ch);

int main() {
    char s[21];
    cin >> s;
    for (int i = 0; i < strlen(s); i++) {
        if (isVowel(s[i])) {
            s[i] = s[i]-32;
        }
    }
    cout << s;
    return 0;
}

int isVowel(char ch) {
    if(strchr("aeiou", ch)) {
        return 1;
    }
    return 0;
}

```

4. <https://www.pbinfo.ro/probleme/1866/prosir>

- Date intrare: "ana are multe mare si o gutuie."
- Date iesire: "an5 ar5 mult5 mar5 s5 5 gutui5."
- Solutie:

```

#include <iostream>
#include <fstream>
#include <cstring>

using namespace std;

int main() {
    ifstream fin("prosir.in");
    ofstream fout("prosir.out");
    char line[201];
    fin.getline(line, sizeof(line));
    int lineLength = strlen(line);
    for(int i = 0; i < lineLength;i++) {
        if(line[i+1] == ' ' && isalnum(line[i])) {
            line[i] = '5';
        }
    }
    line[lineLength-2] = '5';
    fout.write(line, lineLength);
}

```

```
    fin.close();  
    fout.close();  
    return 0;  
}
```

5. <https://www.pbinfo.ro/probleme/1456/cuvant>

- Date intrare 1: "inscriptibil"
- Date iesire 1 : "DA"
- Date intrare 2: "iii"
- Date iesire 2 : "NU"
- Solutie:

```
#include <iostream>  
#include <cstring>  
  
using namespace std;  
  
int isConsonant(char ch);  
int isVowel(char ch);  
  
int main() {  
    char word[101];  
    cin.getline(word, 101);  
    int hasConsonants = 0;  
    int hasOnlyI = 1;  
    for(int i = 0; i < strlen(word); i++) {  
        if(strchr("aeou", word[i])) {  
            hasOnlyI = 0;  
            break;  
        } else if(isConsonant(word[i])){  
            hasConsonants = 1;  
        }  
    }  
    if (hasConsonants && hasOnlyI) {  
        cout << "DA";  
    } else {  
        cout << "NU";  
    }  
    return 0;  
}  
  
int isConsonant(char ch) {  
    return ch > 'a' && ch <= 'z' && !isVowel(ch);  
}  
  
int isVowel(char ch) {  
    if(strchr("aeiou", ch)) {  
        return 1;  
    }  
}
```

```
    }  
    return 0;  
}
```

6. <https://www.pbinfo.ro/probleme/2828/acronim>

- Date intrare: "Universitatea de Arte Plastice BUCURESTI"
- Date iesire: "UAPB"
- Solutie:

```
#include <iostream>  
#include <cstring>  
#include <cctype>  
  
using namespace std;  
  
int main()  
{  
    char word[101];  
    cin.getline(word, 101);  
  
    char* pch = strtok(word, " ");  
    while (pch != NULL) {  
        if (isupper(pch[0])) {  
            cout << pch[0];  
        }  
        pch = strtok(NULL, " ");  
    }  
  
    return 0;  
}
```

7. <https://www.pbinfo.ro/probleme/890/nrvocale>

- Date intrare: "Ana are 5 mere si trei nuci"
- Date iesire: "E"
- Solutie:

```
#include <iostream>  
#include <cstring>  
#include <cctype>  
  
using namespace std;  
  
int isVowel(char);  
  
int main()  
{  
    char word[256];
```

```
cin.getline(word, 256);
char vowels[5] = { 'a', 'e', 'i', 'o', 'u' };
int vowelsFrequency[5] = { 0 };

for (int i = 0; i < strlen(word); i++) {
    if (isVowel(word[i])) {
        char vowel = tolower(word[i]);
        switch (vowel) {
            case 'a':
                vowelsFrequency[0]++;
                break;
            case 'e':
                vowelsFrequency[1]++;
                break;
            case 'i':
                vowelsFrequency[2]++;
                break;
            case 'o':
                vowelsFrequency[3]++;
                break;
            default:
                vowelsFrequency[4]++;
        }
    }
}

int maxAparitii = 0;
for (int i = 0; i < 5; i++) {
    if (vowelsFrequency[i] > maxAparitii) {
        maxAparitii = vowelsFrequency[i];
    }
}

for (int i = 0; i < 5; i++) {
    if (vowelsFrequency[i] == maxAparitii) {
        cout << (char)toupper(vowels[i]);
        break;
    }
}

return 0;
}

int isVowel(char ch) {
    if (strchr("AEIOUaeiou", ch)) {
        return 1;
    }
    return 0;
}
```

- Date intrare: "program"
- Date iesire:

```

program
progra
progr
prog
pro
pr
p
program
rogram
ogram
gram
ram
am
m

```

- Solutie:

```

#include <iostream>
#include <cstring>

using namespace std;
int main() {
    char word[10];
    cin >> word;
    for(int i = strlen(word); i >=0; i--) {
        char temp[i];
        strncpy(temp, word, i);
        temp[i]='\0';
        cout << temp;
        if (i != 0) {
            cout << endl;
        }
    }
    for(int i =0; i < strlen(word); i++) {
        char temp[i];
        strcpy(temp, word+i);
        cout << temp << endl;
    }
    return 0;
}

```

9. <https://www.pbinfo.ro/probleme/84/interschimbarelitere>

- Date intrare: "PrograM"
- Date iesire: "PrMgrao"
- Solutie:

```
#include <iostream>
#include <cstring>

int isConsonant(char ch);
int isVowel(char ch);

using namespace std;
int main() {
    char word[10];
    cin >> word;
    int vowelIndex = -1;
    int consonantIndex = -1;

    for(int i = 0; i < strlen(word); i++) {
        if(vowelIndex == -1 && isVowel(word[i])) {
            vowelIndex = i;
        } else if (isConsonant(word[i])) {
            consonantIndex = i;
        }
    }

    if (vowelIndex != -1 && consonantIndex != -1) {
        char temp = word[vowelIndex];
        word[vowelIndex] = word[consonantIndex];
        word[consonantIndex] = temp;
        cout << word;
    } else {
        cout << "IMPOSIBIL";
    }

    return 0;
}

int isVowel(char ch) {
    return strchr("aeiouAEIOU", ch) != NULL;
}

int isConsonant(char ch) {
    if (isalnum(ch) && !isVowel(ch)) {
        return 1;
    } else {
        return 0;
    }
}
```

10. <https://www.pbinfo.ro/probleme/85/inserareasterisc>

- Date intrare: "ana are mere"
- Date iesire: "ana are mere"
- Solutie:



```

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

int isVowel(char ch);

using namespace std;
int main() {
    char word[101];
    char result[201]="";
    int index =0, hasVowels = 0;

    cin.getline(word,101);

    for(int i = 0; i < strlen(word); i++) {
        result[index] = word[i];
        if(isVowel(word[i])) {
            index++;
            result[index] = '*';
            hasVowels =1;
        }
        index++;
    }
    if(hasVowels)
    {
        cout<<result;
    }
    else
    {
        cout<<"FARA VOCALE";
    }

    return 0;
}

int isVowel(char ch) {
    return strchr("aeiou", ch) != NULL;
}

```

## Recapitulare teorie structuri de date

O structură de date este o colecție de informații grupate sub un singur nume. Aceste informații, cunoscute drept membri, pot avea tipuri de date diferite și lungimi diferite. Structurile de date pot fi definite în C++ folosind următoarea sintaxă:

```

struct nume_tip {
    tip_data_1 id_membru_1;
    tip_data_2 id_membru_2;
    tip_data_3 id_membru_3;
    .

```

```
.  
} denumiri_obiecte;
```

unde **nume\_tip** este o denumire pentru tipul structură, **denumiri\_obiecte** poate fi un set de identificatori valizi de obiecte care sunt de acel tip structură. Între acolade **{}** avem o listă cu informațiile membre, fiecare dintre ele fiind precizată cu tip de dată și un identificator. De exemplu:

```
struct produs {  
    int greutate;  
    double pret;  
} ;  
  
produs mar;  
produs banana, pepene;
```

Avem o definiție de tip structură, denumit **produs**, care are doi membri: **greutate** și **pret**, fiecare de un alt tip de dată fundamental. Această definiție creează un nou tip (**produs**), care este apoi folosit pentru a declara trei obiecte (**variabile**) de acest tip: **mar**, **banana** și **pepene**. Să observăm că, odată definit, tipul produs poate fi folosit la fel ca orice alt tip de dată fundamental.

Chiar la sfârșitul definiției struct, înainte de simbolul (**;**), câmpul opțional **denumiri\_obiecte** poate fi folosit pentru a declara mai direct obiecte de acel tip structură. De exemplu, obiectele structură **mar**, **banana** și **pepene** pot fi declarate simultan cu definiția tipului de dată structură:

```
struct produs {  
    int greutate;  
    double pret;  
} mar, banana, pepene;
```

În acest caz, când precizăm **denumiri\_obiecte**, numele tipului de dată (**produs**) devine opțional:

- **NOTA:** **struct** necesită fie un **nume\_tip** fie cel puțin o **denumire\_obiect**, dar nu neapărat amândouă simultan.
- **NOTA 2:** Este foarte important să înțelegem diferența dintre numele tipului de dată (**produs**) și un obiect de acest tip (**mar**, **banana** și **pepene**). Putem declara mai multe obiecte (ca **mar**, **banana** și **pepene**) cu un singur tip structură (**produs**).

O dată declarate cele trei obiecte de un anumit tip (**mar**, **banana** și **pepene**), putem accesa informațiile membru ale lor direct. Sintaxa pentru aceasta presupune inserarea simbolului punct (**.**) între denumirea obiectului și identificatorul membrului. De exemplu, am putea opera cu oricare dintre aceste elemente ca și cum ele ar fi variabile obișnuite declarate cu propriul lor tip de dată:

```
mar.greutate  
mar.pret
```

```
banana.greutate  
banana.pret  
pepene.greutate  
pepene.pret
```

Fiecare dintre ele are tipul de dată corespunzător membrului la care se referă: `mar.greutate`, `banana.greutate` și `pepene.greutate` sunt de tip `int`, în timp ce `mar.pret`, `banana.pret` și `pepene.pret` sunt de tip `double`.

Avem aici un exemplu real de folosire a tipului structură în care salvăm date despre filmul nostru favorit. Putem observa cum putem initializa un membru, direct dar și de la tastatură:

```
#include <iostream>  
#include <string.h>  
using namespace std;  
  
struct FilmFavorit{  
    char titlu[51];  
    int anLansare;  
} al_meu, al_tau;  
  
void afisareFilm (FilmFavorit film);  
  
int main ()  
{  
  
    strcpy(al_meu.titlu, "Lord of the Rings: Return of the King\0");  
    al_meu.anLansare = 2003;  
  
    cout << "Tasteaza titlul filmului tau favorit: ";  
    cin.getline(al_tau.titlu, 51);  
    cout << "Tasteaza anul in care a fost lansat: ";  
    cin >> al_tau.anLansare;  
  
    cout << "Filmul meu favorit: " << endl;  
    afisareFilm(al_meu);  
    cout << "Filmul tau favorit: " << endl;  
    afisareFilm(al_tau);  
    return 0;  
}  
  
void afisareFilm (FilmFavorit film)  
{  
    cout << "Titlu: " << film.titlu << " si a fost lansat in anul: " <<  
    film.anLansare << endl;  
}
```

Pentru final, o să vedem un exemplu în care citim de la tastatură un vector în care fiecare element, reprezintă o variabilă de tipul `FilmFavorit`:

```
#include <iostream>
using namespace std;

struct FilmFavorit{
    char titlu[51];
    int anLansare;
};

void afisareFilm (FilmFavorit film);

int main ()
{
    int n;
    cout << "Cate filme doriti sa introduceti: ";
    cin >> n;
    FilmFavorit filmeFavorite[n];

    for (int i = 0; i < n; i++) {
        cout << "Introduceti titlul pentru filmul #" << i+1 << " (max 51 caractere):" << endl;
        cin.ignore();
        cin.getline(filmeFavorite[i].titlu, 51);
        cout << "Introduceti anul in care filmul a fost lansat: " << endl;
        cin >> filmeFavorite[i].anLansare;
    }

    cout << "Ati introdus urmatoarele filme: " << endl;
    for (int i = 0; i < n; i++) {
        cout << "#" << i+1<<": " << endl;
        afisareFilm(filmeFavorite[i]);
        cout << endl;
    }
    return 0;
}

void afisareFilm (FilmFavorit film)
{
    cout << "Titlu: '" << film.titlu << "' lansat in anul: " << film.anLansare
<< endl;
}
```

- NOTA: Daca s-a inteles tot ce s-a scris mai sus, problemele de la bacalaureat, pe acest topic, nu au cum sa puna vreo dificultate.