

Sesiunea 12

Agenda

- Rezolvare subiectul III
- Exerciții matrici oarecare

Rezolvare subiectul III

- 1
 - Solutie:

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

int estePNumar(int n);
int kpn(int a, int b, int k);

int main() {
    cout << kpn(27, 50, 3);
    return 0;
}

int kpn(int a, int b, int k){
    int contor = 0;
    int rezultat = -1;
    for(int i = a; i<=b; i++){
        if (estePNumar(i)){
            contor++;
            if (contor == k){
                rezultat = i;
                break;
            }
        }
    }
    return rezultat;
}

int estePNumar(int n) {
    int sumaDivizori = 0;
    for (int i = 1; i <= n; i++) {
        if (n % i == 0) {
            sumaDivizori+= i;
        }
    }

    if (sumaDivizori % 2 == n % 2) {
        return 1;
    } else {
```

```
        return 0;
    }
}
```

- 2

- Solutie:

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

using namespace std;

int main() {
    char text[101], rezultat[101];
    // Ne asiguram ca nu o sa citim garbage din memorie
    text[0] = '\0';
    rezultat[0] = '\0';
    cin.getline(text, 101);
    int contor = 0;
    char *cuvant = strtok(text, " ");
    while (cuvant != NULL) {
        if (strlen(cuvant) % 2 == 0) {
            // il copiem direct in rezultat
            strcat(rezultat, cuvant);
            strcat(rezultat, " ");
        } else {
            int lungimeCuvant = strlen(cuvant);
            char oglindit[lungimeCuvant+1];
            for(int i = 0; i < lungimeCuvant; i++) {
                oglindit[i] = cuvant[lungimeCuvant - 1 - i];
            }
            oglindit[lungimeCuvant] = '\0';
            if (strcmp(oglindit, cuvant) != 0) {
                contor++;
            }
            strcat(rezultat, oglindit);
            strcat(rezultat, " ");
        }
        cuvant = strtok(NULL, " ");
    }
    if (contor == 0) {
        cout << "nu exista";
    } else {
        cout << rezultat;
    }
    return 0;
}
```

- 3

- Explicatie

O sa scriem un algoritm in care vom cauta cel mai mic si cel mai mare numar de doua cifre. In cazul in care nu vom gasi cel putin un astfel de numar, se va afisa "NU EXISTA". Algoritmul va fi eficient din punct de vedere al timpului de executie deoarece se va face o singura parcurgere a numerelor din fisier. In acelasi timp, algoritmul este eficient din punct de vedere al memoriei utilizate deoarece nu se foloseste alte structuri de date pentru stocarea si manipularea numerelor, practic, din maximum de 10^6 numere cate pot fi in fisier, noi in memorie o sa avem maximum 3, in acelasi moment: min, max si numarul curent pe care il citim.

- Solutie:

```
#include <iostream>
#include <fstream>

using namespace std;

int main() {
    ifstream fin("bac.txt");

    int minimDouaCifra=-1;
    int maximDouaCifre=-1;
    int numar;
    while(fin >> numar) {
        // aici verificam daca are doua cifre
        if (numar >= 10 && numar <= 99) {
            // daca suntem la inceput, initializam ambele cu
            // aceeasi valoare
            if (minimDouaCifra == -1 && maximDouaCifre == -1) {
                minimDouaCifra = numar;
                maximDouaCifre = numar;
            } else {
                // altfel updatam cele doua numere, min, max
                if (numar < minimDouaCifra) {
                    minimDouaCifra = numar;
                } else if (numar > maximDouaCifre) {
                    maximDouaCifre = numar;
                }
            }
        }
    }

    // aici putem lua oricare dintre minime
    // daca valoarea oricareia este -1 inseamna ca nu am gasit un
    // numar cu 2 cifre
    if (minimDouaCifra == -1) {
```

```

        cout << "nu exista";
    } else {
        cout << minimDouaCifra-1 << " " << maximDouaCifre +1;
    }

    fin.close();
    return 0;
}

```

Exercitii matrici oarecare

1.
 - Link: <https://www.pbinfo.ro/probleme/767/sumapare2>
 - Solutie:

```

#include <iostream>
#include <fstream>

using namespace std;

int main() {
    int n,m;

    cin >> n >> m;
    int matrice[n][m];
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            cin >> matrice[i][j];
        }
    }
    int suma = 0;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            if(matrice[i][j] % 2 == 0) {
                suma += matrice[i][j];
            }
        }
    }

    cout << suma;
    return 0;
}

```

2.
 - Link: <https://www.pbinfo.ro/probleme/658/sumalinii>
 - Solutie:

```

#include <iostream>
#include <fstream>

using namespace std;

```

```

int main() {
    int n,m;

    cin >> n >> m;
    int matrice[n][m];
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            cin >> matrice[i][j];
        }
    }
    for (int i = 0; i < n; i++) {
        int sumaLinie = 0;
        for (int j = 0; j < m; j++) {
            sumaLinie += matrice[i][j];
        }
        cout << sumaLinie << " ";
    }

    return 0;
}

```

3.
 - Link: <https://www.pbinfo.ro/probleme/666/nrprime>
 - Solutie:

```

#include <iostream>
#include <fstream>

using namespace std;
int estePrim(int n);

int main() {
    int n,m;
    int contor = 0;

    cin >> n >> m;
    int matrice[n][m];
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            cin >> matrice[i][j];
        }
    }
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            if ((i+1) % 2 == 0 && estePrim(matrice[i][j])) {
                contor++;
            }
        }
    }
    cout << contor;
}

```

```

        return 0;
    }

    int estePrim(int n) {
        if (n < 2) {
            return 0;
        }
        int rezultat = 1;
        for (int i = 2; i*i <= n; i++) {
            if (n % i == 0) {
                rezultat = 0;
                break;
            }
        }
        return rezultat;
    }
}

```

4. ◦ Link: <https://www.pbinfo.ro/probleme/2807/matrice10>
 ◦ Solutie:

```

#include <iostream>

using namespace std;

int main() {
    int n,m;

    cin >> n >> m;
    int matrice[n][m];
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            cin >> matrice[i][j];
        }
    }
    int indexLinieMinim1 = 0, indexLinieMinim2 = 0;
    for (int i = 0; i < n; i++) {
        if (matrice[i][0] < matrice[indexLinieMinim1][0]) {
            indexLinieMinim1 = i;
        }
        if (matrice[i][m-1] < matrice[indexLinieMinim2][m-1]) {
            indexLinieMinim2 = i;
        }
    }

    int aux = matrice[indexLinieMinim1][0];
    matrice[indexLinieMinim1][0] = matrice[indexLinieMinim2][m-1];
    matrice[indexLinieMinim2][m-1] = aux;

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {

```

```
        cout << matrice[i][j] << " ";
    }
    cout << endl;
}

return 0;
}
```

5. ◦ Link: <https://www.pbinfo.ro/probleme/2825/chenar2>
 ◦ Solutie:

```
#include <iostream>

using namespace std;

int main() {
    int n,m;

    cin >> n >> m;
    int matrice[n][m];
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            cin >> matrice[i][j];
        }
    }

    int x;
    cin >> x;
    int gasit = 0;

    // cautam pe prima linie
    for (int i = 0; i < m; i++) {
        if (matrice[0][i] == x) {
            gasit = 1;
            break;
        }
    }
    if (!gasit) {
        // cautam pe ultima coloana
        for (int i = 0; i < n; i++) {
            if (matrice[i][m-1] == x) {
                gasit = 1;
                break;
            }
        }
    }

    if (!gasit) {
        // cautam pe ultima linie
        for (int i = 0; i < m; i++) {
```

```
        if (matrice[n-1][i] == x) {
            gasit = 1;
            break;
        }
    }
}

if (!gasit) {
    // cautam pe prima coloana
    for (int i = 0; i < n; i++) {
        if (matrice[i][0] == x) {
            gasit = 1;
            break;
        }
    }
}

if (gasit) {
    cout << "DA";
} else {
    cout << "NU";
}

return 0;
}
```

Exercitii propuse:

1. Subiectul III, Exercițiul 2 din Bac 2023 Iunie
2. Subiectul III, Exercițiul 2 din Bac 2022 August
3. Subiectul III, Exercițiul 2 din Bac 2022 Speciala
4. Subiectul III, Exercițiul 2 din Bac 2021 August