

# Rezolvare model subiect bacalaureat informatica 2023

---

## Adresa:

- [https://www.pbinfo.ro/resurse/9dc152/examene/2023/E\\_d\\_Informatica\\_2023\\_sp\\_MI\\_C\\_var\\_model.pdf](https://www.pbinfo.ro/resurse/9dc152/examene/2023/E_d_Informatica_2023_sp_MI_C_var_model.pdf)
- [https://www.pbinfo.ro/resurse/9dc152/examene/2023/E\\_d\\_Informatica\\_2023\\_sp\\_MI\\_bar\\_model.pdf](https://www.pbinfo.ro/resurse/9dc152/examene/2023/E_d_Informatica_2023_sp_MI_bar_model.pdf)

## Rezolvare

### Subiectul I

1. b
2. d
3. a
4. c
5. Detalii:
  - Se numește ciclu eulerian un ciclu care conține toate muchiile grafului.
  - Se numește ciclu hamiltonian un ciclu elementar care conține toate vârfurile grafului
  - Avem: ciclu eulerian lungime 11 (11 muchii) și ciclu hamiltonian de lungime 7. => 6 noduri
  - După desen, rezulta 5 pot fi sterse => D

### Subiectul II

1.
  - a: 90
  - b: Oricare doua numere din intervalul [70, 74]
  - c:

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

int main()
{
    int m, n, p, q, s = 0;
    cin >> m >> n >> p >> q;
    for (int x = p; x <=q; x++) {
        if (x % m == 0 || x % n == 0) {
            s = s+x;
        }

        if (x % m == 0 && x % n == 0) {
            s = s-x;
        }
    }
    cout << s;
}
```

- o d

```

citeste m,n,p,q (numere naturale nenule, p<= q)
s <- 0
x<-p
cat timp (x <= q) executa:
    daca x%m=0 sau x%n=0 atunci
        s <- s+x;
    daca x%m=0 si x%n=0 atunci
        s <- s-x
    x++;
scrie s;

```

2. Exemplu cu tot cu sortare + verificare:

```

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

struct echipa {
    char nume[50];
    int rezultat;
};

int main()
{
    struct echipa c[3];
    strcpy(c[0].nume, "Bayern");
    c[0].rezultat = 100;

    strcpy(c[1].nume, "Inter");
    c[1].rezultat = 250;

    strcpy(c[2].nume, "Milan");
    c[2].rezultat = 1000;

    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 2; j++) {
            if (c[j].rezultat < c[j+1].rezultat) {
                echipa temp = c[j];
                c[j] = c[j+1];
                c[j+1] = temp;
            }
        }
    }

    for (int i = 0; i < 3; i++) {
        cout << c[i].nume << " ";
    }
}

```

```
}  
}
```

3. Se va afisa:

```
#include <iostream>  
#include <string.h>  
  
using namespace std;  
  
int main() {  
    char s1[31], s2[31];  
    int p;  
    strcpy(s1, "plantau fistic");  
    p = strchr(s1, ' ')-s1; //p = 7  
    strcpy(s2, s1+p+1); //s2 va contine "fistic"  
    strcpy(s1+p-1, s2+2); //s1 = "plantastic"  
    strcpy(s2+1, s1+2); //s2="fantastic"  
    cout << p << s2; // 7fantastic  
}
```

### Subiectul III

1. Solutie:

```
#include <iostream>  
#include <string.h>  
  
using namespace std;  
  
int DoiTrei(int n);  
  
int main() {  
    int n;  
    cin >> n;  
    cout << DoiTrei(n);  
}  
  
int DoiTrei(int n) {  
    int rezultat = 1;  
    while (n > 0) {  
        int ultimaCifra = n%10;  
        if (ultimaCifra != 2 && ultimaCifra != 3) {  
            rezultat = 0;  
            break;  
        }  
        n = n/10;  
    }  
}
```

```

        return result;
    }

```

## 2. Solutie:

```

#include <iostream>

using namespace std;

int main() {
    int n;
    cin >> n;
    int matrice[6][6] = {
        {2,0,0,2,1,3},
        {3,1,3,1,2,0},
        {2,1,3,3,2,0},
        {0,2,1,3,1,1},
        {3,1,2,0,0,2},
        {0,0,0,2,1,3},
    };
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            cin >> matrice[i][j];
        }
    }
    int existaZone = 0;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            int zonaNord = matrice[i-1][j];
            int zonaEst = matrice[i][j+1];
            int zonaSud = matrice[i+1][j];
            int zonaVest = matrice[i][j-1];
            int codElementCurent = matrice[i][j];
            if (i == 0) {
                if (j == 0) {
                    if (codElementCurent == 3 && zonaEst != 0 &&
zonaSud != 0){
                        existaZone = 1;
                        cout << i+1 << " ";
                    }
                } else if (j == n-1) {
                    if ( codElementCurent == 3 && zonaVest != 0 &&
zonaSud != 0){
                        existaZone = 1;
                        cout << i+1 << " ";
                    }
                } else {
                    if (codElementCurent == 3 && zonaVest != 0 &&
zonaEst != 0 && zonaSud != 0){
                        existaZone = 1;
                        cout << i+1 << " ";
                    }
                }
            }
        }
    }
}

```

```

    }
    }
    } else if (i == n-1) {
        if (j == 0) {
            if (codElementCurent == 3 && zonaNord != 0 &&
zonaEst != 0){
                existaZone = 1;
                cout << i+1 << " ";
            }
        } else if (j == n-1) {
            if (codElementCurent == 3 && zonaVest != 0 &&
zonaNord != 0){
                existaZone = 1;
                cout << i+1 << " ";
            }
        } else {
            if (codElementCurent == 3 && zonaVest != 0 &&
zonaEst != 0 && zonaNord != 0){
                existaZone = 1;
                cout << i+1 << " ";
            }
        }
    } else {
        if (codElementCurent == 3 && zonaNord != 0 && zonaEst
!= 0 && zonaSud!= 0 && zonaVest != 0) {
            existaZone = 1;
            cout << i+1 << " ";
        }
    }
}
}
if (!existaZone) {
    cout << "nu exista";
}
}

```

### 3. Solutie:

- In limbaj natural:

Citim primele n numere care reprezinta x-urile, dupa care, citim y-urile unul cate unul.

Imediat ce gasim o pereche ce nu corespunde cerintei, intrerupem citirea restului de numere.

Astfel, eficienta programului se datoreaza faptului ca evitam citirea intregului fisier, imediat cum am gasit un element (y) ce invalideaza

- Cod:

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

using namespace std;

int main() {
    int n;
    int p0rdonate = 1;
    ifstream fin("bac.txt");

    fin >> n;
    int xPairs[n];

    for (int i = 0; i < n; i++) {
        fin >> xPairs[i];
    }

    for (int i = 0; i < n; i++) {
        int yPairElement;
        fin >> yPairElement;
        if ( (xPairs[i] % 2) == (yPairElement % 2)) {
            continue;
        } else {
            if (xPairs[i] <= yPairElement) {
                p0rdonate = 0;
                break;
            }
        }
    }

    if (p0rdonate == 1) {
        cout << "DA";
    } else {
        cout << "NU";
    }
    fin.close();
}
```