

# Varianta culeasa din culegerea bacalaureat la informatica - Teste Rezolvate (2023) V1

---

## Subiectul 1.

1. b
2. a
3. c
  - numerele sunt: 2125; 2135; 2145; 2225; 2235; 2245; 2325; 2335; 2345; 2425; 2435; 2445; 2525; 2535; 2545
4. c - Formula este  $2 \cdot \frac{n \cdot (n-1)}{2}$
5. d (deoarece ciclic inseamna sa ai lant simplu insa noi avem un lant elementar)

## Subiectul II

1.
  - a: 1355
  - b: Programul elimina toate cifrele impare si va returna numarul format din cifrele pare ale acestuia incrementate cu 1.
    - Raspuns: 16, 36, 56, 76, 96, 61, 63, 65, 67, 69
  - c
    - Raspuns:

```
citeste n (numar natural)
z <- 0; p<-1;
daca n > 0 executa
    executa
        c <- n%10; n <- n/10
        daca c%2 = 0 atunci
            z <- z + p * (c+1); p <- p * 10
        cat timp n > 0

scrie z
```

- d

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{

    int n, z = 0, p = 1;
    cin >> n;
    while (n > 0) {
        int c = n % 10;
```

```
        n = n / 10;
        if (c % 2 == 0) {
            z = z + p * (c+1);
            p = p * 10;
        }
    }
    cout << z;
    return 0;
}
```

2. Raspuns: Vom folosi distanta dintre 2 puncte

```
float distantaA0 = sqrt((A.x - 0) * (A.x - 0) + (A.y - 0) * (A.y - 0));
float distantaB0 = sqrt((B.x - 0) * (B.x - 0) + (B.y - 0) * (B.y - 0));
distantaA0 == distantaB0;
```

3. Raspuns:

```
using namespace std;
int main()
{
    char s[31];
    int i;
    strcpy(s, "bac2723cunota37");
    for (int i = 0; i < strlen(s); i++) {
        if (s[i] == '7') {
            cout << '0';
        } else if (s[i] == '3') {
            cout << '1';
        } else {
            cout << s[i];
        }
    }
    return 0;
}
```

## Subiectul III

1. Raspuns:

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

```

void sub(int a, int b, int &n, int v[]);

int main() {
    int n = 10, v[] = {3551, 149, 3798, 502, 75, 2515, 51, 151, 489, 653},
    a = 5, b = 2;
    sub(a, b, n, v);
    cout << endl << "n = " << n << endl;
    cout << "v = ";
    for (int i = 0; i < n; i++) {
        cout << v[i] << " ";
    }
    return 0;
}

void sub(int a, int b, int &n, int v[]) {
    int contor = 0;
    int i = 0;
    while (i < n) {
        int numar = v[i];
        int areA = 0;
        int areB = 0;
        while(numar > 0) {
            int ultimaCifra = numar % 10;
            numar = numar / 10;
            if (ultimaCifra == a) {
                areA = 1;
            } else if (ultimaCifra == b) {
                areB = 1;
                break;
            }
        }
        if (areA && !areB) {
            for (int j = i; j < n; j++) {
                v[j] = v[j+1];
            }
            contor++;
            n--;
        } else {
            i++;
        }
    }
    n = contor;
}

```

## 2. Rezolvare:

```

#include <iostream>
using namespace std;

int main() {
    int n = 3, m = 5;
    int matrice[n][m];
    for (int i = 0; i < n; i++) {

```

```
        for (int j = 0; j < m; j++) {
            matrice[i][j] = i + j + 1;
        }
    }

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

    return 0;
}
```

3.   ◦ Explicatie eficienta:
- Programul este eficient din punct de vedere al memoriei deoarece tinem in memorie doar cate un singur numar din fisier, impreuna cu doua contoare. Din punct de vedere al timpului, programul este eficient deoarece parcurgem o singura data fisierul text.
- Solutie:

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

int main() {
    ifstream fin ("bac.in");
    int numarCitit;
    int lungimeMaxima = 0;
    int lungimeCurenta = 0;
    while (fin >> numarCitit){
        if (numarCitit % 2 == 0) {
            lungimeCurenta++;
        } else {
            if (lungimeCurenta > lungimeMaxima) {
                lungimeMaxima = lungimeCurenta;
            }
            lungimeCurenta = 0;
        }
    }
    cout << lungimeMaxima;

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