

In aceasta sesiune vom rezolva subiectele ce au fost date la bacalaureat 2022

Subiectul I

1. c
2. d
3. a
4. b
5. c

Subiectul II

1.
 - a. 4
 - b. 11, 13
 - c.

```
#include <iostream>

using namespace std;
int main() {
    int n, i = 2, k = 0;
    cin >> n;
    while ( n >= i) {
        while (n % i == 0) {
            k = k + 1;
            n = n / i;
        }
        if (i == 2) {
            i = i + 1;
        } else {
            i = i + 2;
        }
    }
    cout << k;
    return 0;
}
```

d.

```
citeste n
i <- 2; k <-0
cat timp n>= i executa
    daca n % i = 0 atunci
```

```

    repeta
        k <- k+1
        n <- [n/i]
        pana cand n % i != 0
        daca i = 2 atunci i <- i + 1
        altfel i <- i + 2
    scrie k

```

2. Oricare 2 numere dintre urmatoarele: 2022, 2023, 2024, 2025

3.

- Solutie 1:

```

char s[51], char id[51];
// pch va contine prenumele (e.g Ana)
char* pch = strtok(s, " ");
// pch va contine numele (e.g Popescu)
pch = strtok(NULL, " ");
strcpy(id, pch);
strcat(id, "2022");

```

- Solutie 2:

```

char s[51], char id[51];
// Facem +1 ca sa nu copiem si spatiul ce separa numele
strcpy(id, strchr(s, ' ')+1);
strcat(id, "2022");

```

Subiectul III

1. Solutie:

```

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

void secventa(int& n);

int main()
{
    int n = 202233228;
    secventa(n);
    cout << n;
}

void secventa(int& n) {

```

```

int result =0;
int pozitii = 0;
while (n > 0) {
    int ultimaCifra = n % 10;
    n = n / 10;
    int urmatoareaCifra = n % 10;
    if (ultimaCifra == 2 && urmatoareaCifra == 2) {
        result = 20 * pow(10, pozitii) + result;
        n /= 10;
        pozitii = pozitii + 2;
    } else {
        result = ultimaCifra * pow(10, pozitii++) + result;
    }
}

n = result;
}

```

2. Solutie:

```

#include <iostream>

int getMinimumValue(int a, int b, int c, int d);
int getMinimumValue(int a, int b, int c);
int getMinimumValue(int a, int b);

using namespace std;

int main() {
    int m = 5, n = 4;
    cin >> m >> n;
    int matrice[m][n];

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

    int nisipAaugat = 0;
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            int minValue;
            int parcelaNord = matrice[i-1][j];
            int parcelaEst = matrice[i][j + 1];
            int parcelaSud = matrice[i + 1][j];
            int parcelaVest = matrice[i][j-1];
            if (i == 0) {
                if (j == 0) {
                    minValue = getMinimumValue(parcelaEst, parcelaSud);
                } else if (j == n-1) {

```

```

        minValue = getMinimumValue(parcelaSud, parcelaVest);
    } else {
        minValue = getMinimumValue(parcelaSud, parcelaVest,
parcelaEst);
    }
    } else if (i == m-1) {
        if (j == 0) {
            minValue = getMinimumValue(parcelaNord, parcelaEst);
        } else if (j == n-1) {
            minValue = getMinimumValue(parcelaNord, parcelaEst);
        } else {
            minValue = getMinimumValue(parcelaNord, parcelaVest,
parcelaEst);
        }
    } else {
        minValue = getMinimumValue(parcelaNord, parcelaEst,
parcelaVest, parcelaSud);
    }

    if (matrice[i][j] < minValue) {
        int diferenta = minValue - matrice[i][j];
        matrice[i][j] = diferenta;
        nisipAaugat += diferenta;
    }
}

cout << "S-au adaugat extra: " << nisipAaugat << " metri cubi de
nisip.";
}

int getMinimumValue(int a, int b) {
    if (a < b) {
        return a;
    } else {
        return b;
    }
}

int getMinimumValue(int a, int b, int c) {
    int minBC = getMinimumValue(b, c);
    return getMinimumValue(a, minBC);
}

int getMinimumValue(int a, int b, int c, int d) {
    int min = a;
    if (b < min) min = b;
    if (c < min) min = c;
    if (d < min) min = d;
    return min;
}

```

3.

- a. In limbaj natural:
 - Citim primele doua numere care reprezinta intervalul in care cautam numerele.
 - Dupa care citim numerele de pe linia urmatoare, unul cate unul
 - Initializam o variabila care va reprezenta ultimul numar valid citit, cu -1.
 - Un numar este valid daca: - Este mai mare sau egal cu x - Este mai mic sau egal cu y - Este mai mare strict decat ultimul numar valid citit
 - Initializam un contor ce va contoriza numerele valide.
 - In cazul in care, ajungem la un numar care este mai mare decat y, stim ca nu se mai respecta conditia din enunt si astfel putem incheia contorizarea.
 - Programul este eficient deoarece
 1. Parcurgem cel mult o data fisierul
 2. Stiind ca numerele sunt ordonate crescatoare, in cazul in care intalnim un numar in afara intervalului, am oprit contorizarea.
- b. Solutie

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

using namespace std;

int main()
{
    int x,y;
    ifstream fin("bac.txt");
    fin >> x >> y;
    int lastValidDigit = -1;
    int counter = 0;
    while(!fin.eof()) {
        int number;
        fin >> number;
        if (number >= x && number > lastValidDigit && number <= y) {
            counter++;
            lastValidDigit = number;
        } else if (number > y) {
            break;
        }
    }
    cout << counter;
    fin.close();
}
```

Tema pentru data viitoare

- Varianta 2 de pe site-ul modinfo(<https://www.modinfo.ro/bac/variante-test-2021/info/v2.pdf>)
- Primele 10 probleme, pbinfo de aici: https://www.pbinfo.ro/?pagina=probleme-lista&disciplina=0&clasa=10&tag=10&subtag=11&dificultate=1&folosesc_consola=-1&eticheta=

pentru imbunatatirea cunostintelor in a folosi functiile din string.h (strcpy, strcat, etc) IMPERIOS
NECESAR PENTRU BAC