

## BAREM – VARIANTA II

<b>OFICIU.....</b>	<b>10 puncte</b>
<b>SUBIECTUL I .....</b>	<b>35 puncte</b>
1. 1. Degustare de ciocolată.....	20 puncte
• V1: determinarea corectă a valorii $nr$ (cu formula $nr = n - n/cmmdc(n, k)$ .....	20 puncte
• V2: determinarea corectă a valorii $nr$ (simulare, listă circulară) .....	10 puncte
2. Reducere .....	15 puncte
– parcurgerea în paralel a celor două șiruri .....	4 puncte
– calculul sumelor parțiale cu elemente din șirul $a$ și verificarea egalității cu elementul curent din șirul $b$ .....	4 puncte
– identificarea subsecvenței (din șirul $a$ ) de lungime maximă .....	4 puncte
– test dacă suma parțială curentă depășește elementul curent din $b$ .....	1 punct
– test dacă s-a terminat parcurgerea elementelor din șirul $a$ și suma parțială curentă depășește elementul curent din șirul $b$ .....	1 punct
– test dacă, inițial, șirul $b$ are mai multe elemente decât șirul $a$ .....	1 punct
<b>SUBIECTUL II.....</b>	<b>15 puncte</b>
<b>Numărul format din cifrele pare și numărul format din cifrele impare ale numărului <math>n</math></b>	
<b>Cerința a.</b>	
– enunț problemă.....	5 puncte
<b>Cerința b.</b>	
– rezultat calculat corect ( $p = 2060, i = 59$ ).....	3 puncte
<b>Cerința c.</b>	
– algoritm .....	7 puncte
<b>SUBIECTUL III Prelucrări imagine .....</b>	<b>40 puncte</b>
<b>Subprograme:</b>	
a. Citirea datelor de intrare de la tastatură.....	4 puncte
• citirea imaginii.....	2 puncte
• citirea secvenței de transformări.....	2 puncte
b. inversarea unei imagini .....	4 puncte
c. rotirea cu 90 de grade a unei imagini .....	8 puncte
d. aplicarea operației de zoom pe o imagine .....	8 puncte
e. afișarea unei imagini .....	2 puncte
<b>Program principal .....</b>	<b>3 puncte</b>
– determinarea rezultaului .....	3 puncte
– comunicare prin parametri: (signatura subalgorimilor și apelul corect).....	5 puncte
– lizibilitate:	
• comentarii .....	1 punct
• indentare .....	1 punct
• denumiri sugestive .....	1 punct

## 0 solutie posibila:

```
#include <iostream>
using namespace std;
/*****
Subiectul I.1. Degustare de ciocolata
*****/
//calculeaza si returneaza cmmdc a 2 numere naturale a si b
int cmmdc(int a, int b){
    if ((a == b) && (a != 0))
        return 1;
    if (a * b == 0)
        return a + b;
    while (b != 0){
        int c = b;
        b = a % b;
        a = c;
    } //while
    return a;
}
//determina si returneaza nr de copii care nu primesc ciocolata dintre cei n copii
//numarand din k in k. Putem să considerăm număratoarea în cerc ca o număratoare
//liniară în mai multe siruri mici, fiecare cu n copii, obținând un sir mare cu
//p copii (p fiind multiplu de n). Număratoarea se termina atunci când al n-lea
//copil (dintr-un sir mic) primește ciocolata (astfel, urmatorul copil care ar
//trebui să primească ciocolata va fi un al k-lea copil din urmatorul sir mic),
//deci p trebuie să fie și multiplu de k. Asadar, p = cmmmc(n, k). Dintre cei
//p copii, au primit ciocolată exact p / k copii, deci copiii fără ciocolata sunt
//în număr de
//nr = n - p/k = n - cmmmc(n,k)/k = n - (n*k/cmmdc(n,k))/k = n - n/cmmdc(n,k)
int degustareCiocolata(int n, int k){
    return n - n / cmmdc(n, k);
}

/*****
Subiectul I.2. Reducere
*****/
//verifica dacă sirul a cu n numere naturale se poate reduce la sirul b cu m numere
//naturale și determină poziția k din sirul b care reține elementul obținut prin
//însurubarea elementelor din cea mai lungă subsecvență (de lungime nrMax) din sirul a
bool seReduce(int n, int a[], int m, int b[], int &k, int &nrMax){
    if (n < m)
        return false; // sirul a are mai puține elemente decât sirul b
    int i = 0; // index în sirul a
    int j = 0; // index în sirul b
    int sum = 0; // se va încerca obținerea, pe rând, a elementelor din sirul b
    // ca suma de elemente consecutive din sirul a
    int nrInsumate = 0; // numărul elementelor din subsecvența curentă
    nrMax = 0; // numărul maxim de elemente din subsecvențele sirului a
    // ale căror sume sunt egale cu un element din sirul b
    while ((j < m) && (i < n)){ // cât timp nu s-a terminat niciunul dintre siruri
        sum += a[i];
        nrInsumate++;
        if (sum == b[j]){
            if (nrInsumate > nrMax){
                nrMax = nrInsumate; // dacă este cazul, se actualizează nrMax și k
                k = j;
            }
            nrInsumate = 0; // eventual, va urma o subsecvență nouă
            sum = 0;
            j++; i++;
        } //if
        else{
            if (sum < b[j])
                i++;
            else
                return false;
        } //else
    } //while
}
```

```

        if ((i < n) || (j < m))    // daca cel putin unul dintre cele doua siruri nu s-a epuizat
            return false;
        return true;
    }

    /*****
    Subiectul II. Produsul a doua numere
    *****/

    //Subiectul II.a Numărul format din cifrele pare (p) și numărul format din cifrele
    //impare (i) ale numărului n considerate de la stanga la dreapta

    //Subiectul II.b f(205 609, p, i) => p = 2060, i = 59

    //Subiectul II.c
    void cifreIterativ(int n, int & pare, int & impare){
        pare = 0;
        impare = 0;
        int putereImpare = 1;    //putere lui 10 pentru formarea nr cu cifre impare
        int puterePare = 1;    //putere lui 10 pentru formarea nr cu cifre pare
        while (n){
            if (n & 1){    //daca n este impar
                impare = impare + putereImpare * (n % 10);
                n /= 10;
                putereImpare *= 10;
            }    //if
            else{
                pare = pare + puterePare * (n % 10);
                n /= 10;
                puterePare *= 10;
            }    //else
        }    //while
    }

    /*****
    Subiectul III Prelucrare imagine
    *****/
    const int DIMMAX = 320;

    //inverseaza valorile unei matrici binare im cu m linii si m coloane
    void inversare(int m, int im[][DIMMAX]){
        for (int i = 0; i < m; i++){
            for (int j = 0; j < m; j++){
                im[i][j] = 1 - im[i][j];
            }    //for j
        }    //for i
    }

    //roteste o matrice im cu m linii si m coloane
    void rotire(int m, int im[][DIMMAX]){
        int copie[DIMMAX][DIMMAX];
        for (int i = 0; i < m; i++){
            for (int j = 0; j < m; j++){
                copie[i][j] = im[i][j];
            }    //for j
        }    //for i
        for (int i = 0; i < m; i++){
            for (int j = 0; j < m; j++){
                im[j][m - i - 1] = copie[i][j];
            }    //for j
        }    //for i
    }

    //mareste o matrice im cu m linii si m coloane
    void zoom(int &m, int im[][DIMMAX]){
        int imNoua[DIMMAX][DIMMAX];
        int iNou = 0;
        int jNou = 0;

```

```

        for (int i = 0; i < m; i++){
            jNou = 0;
            for (int j = 0; j < m; j++){
                imNoua[iNou][jNou] = im[i][j];
                imNoua[iNou][jNou + 1] = im[i][j];
                imNoua[iNou + 1][jNou++] = im[i][j];
                imNoua[iNou + 1][jNou++] = im[i][j];
            }
            //for j
            iNou += 2;
        }
        //for i
        m *= 2;
        for (int i = 0; i < m; i++){
            for (int j = 0; j < m; j++){
                im[i][j] = imNoua[i][j];
            }
            //for j
        }
        //for i
    }

//afiseaza o matrice im cu m linii si m coloane
void afisare(int m, int im[][DIMMAX]){
    for (int i = 0; i < m; i++){
        for (int j = 0; j < m; j++){
            cout << im[i][j] << " ";
        }
        //for j
        cout << endl;
    }
    //for i
}

//aplica o secventa s de k transformari asupra unei matrici im cu
//m linii si m coloane
void transformari(int k, char s[], int &m, int im[][DIMMAX]){
    for (int i = 0; i < k; i++){
        if (s[i] == 'I')
            inversare(m, im);
        if (s[i] == 'R')
            rotire(m, im);
        if (s[i] == 'Z')
            zoom(m, im);
    }
    //for i
}

//citeste o matrice im cu m linii si m coloane
void citireImagine(int &m, int im[][DIMMAX]){
    cout << "m = "; cin >> m;
    for (int i = 0; i < m; i++){
        for (int j = 0; j < m; j++){
            cin >> im[i][j];
        }
        //for j
    }
    //for i
}

//citeste o secventa s de k transformari
void citireTransformari(int &k, char s[]){
    cout << " k = "; cin >> k;
    for (int i = 0; i < k; i++)
        cin >> s[i];
}

//programul principal
int main(){
    int m = 0;
    int im[DIMMAX][DIMMAX];
    citireImagine(m, im);
    int k = 0;    char s[5];
    citireTransformari(k, s);
    transformari(k, s, m, im);
    afisare(m, im);
    return 0;
}

```