

## MODEL TEST 1

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul efectiv de lucru este de 3 ore.
- În rezolvările cerute, identificatorii utilizați trebuie să respecte precizările din enunț (**bold**), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată).

### Subiectul I

**(20 de puncte)**

Pentru fiecare dintre itemii de la 1 la 5, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 4 puncte.

1. Variabilele **a, b** și **z** sunt întregi, iar  $a \leq b$ . Care dintre expresiile C/C++ următoare are valoarea 1 dacă și numai dacă valoarea variabilei **z** este pară și nu aparține intervalului închis determinat de valorile variabilelor **a** și **b** ?

- a.  $z \% 2 == 0 \ \&\& \ z > a \ || \ z > b$                       b.  $!(z < a \ \&\& \ z > b) \ \&\& \ z \% 2 == 0$   
 c.  $z < a \ \&\& \ z > b \ \&\& \ z \% 2 == 0$                       d.  $!(z >= a \ \&\& \ z <= b) \ \&\& \ z \% 2 == 0$

2. Considerând declarația alăturată, în care câmpurile linia și coloana memorează poziția unei piese pe o tablă de șah, care dintre următoarele expresii are valoarea 1 dacă și numai dacă două piese **p1** și **p2** se află pe aceeași linie sau aceeași coloană?

```
struct { int linia, coloana; } p1, p2;
```

- a.  $p1.linia == p2.linia \ \&\& \ p1.coloana == p2.coloana$   
 b.  $p1.linia == p2.linia \ || \ p1.coloana == p2.coloana$   
 c.  $!(p1.linia != p2.linia \ || \ p1.coloana != p2.coloana)$   
 d.  $p1.coloana == p2.coloana$

3. Se consideră un arbore cu rădăcină cu 10 noduri, numerotate de la 1 la 10, memorat cu ajutorul vectorului de tați  $T = (3, 10, 0, 8, 1, 1, 3, 5, 7, 3)$  care este lungimea maximă a unui lanț care unește rădăcina cu o frunză?

- a. 6                      b. 2                      c. 4                      d. 5

4. Funcția **F** are definiția alăturată. Câte valori egale cu 0 se afișează la apelul **F(5)**?

```
void F(int x)
{   cout << x << ' ';
    if(x != 0) {
        cout << 0 << ' ';
        if (x % 2 == 0)
            F(x-2);
        else
            F(x+1);
    }
}
```

- a. 9                      b. 10                      c. 5                      d. 6

5. Se consideră algoritmul care determină toate permutările distincte de **n** obiecte (numerotate de la 1 la **n**), în care pe orice poziție de rang par se află o valoare pară. De exemplu, pentru **n=5**, primele trei permutări generate în ordine lexicografică sunt: (1,2,3,4,5), (1,2,5,4,3), (1,4,3,2,5). Pentru **n=5**, numărul total de astfel de permutări este:

- a. 12                      b. 10                      c. 9                      d. 7

**Subiectul al II-lea****(40 de puncte)****1. Se consideră algoritmul alăturat, descris în pseudocod.**

S-a notat cu  $x\%y$  restul împărțirii numărului natural  $x$  la numărul natural nenul  $y$  și cu  $[z]$  partea întreagă a numărului real  $z$ .

a) Scrieți valorile care se vor afișa dacă se citesc în ordine numerele **5 15 45 33 81 66 44 87**. (6p.)

b) Dacă pentru  $n, a$  și  $b$  se citesc valorile **5, 50, 100** completați setul de date cu valori care pot fi citite astfel încât, în urma executării algoritmului, valoarea afișată să fie **4**. (6p.)

c) Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)

d) Scrieți în pseudocod un algoritm echivalent cu cel dat în care structura **pentru ...execută**

să fie înlocuită cu o structură repetitivă cu test inițial. (6p.)

**citeste**  $n, a, b$  (numere naturale nenule)

**$m \leftarrow 0$**

**┌ pentru  $i \leftarrow 1, n$  execută**

**| citește  $x$**

**| ┌ dacă  $x \geq a$  și  $x \leq b$  atunci**

**|| ┌ dacă  $x \% 10 = [x/10]$  atunci**

**|| | ┌  $m \leftarrow m + 1$**

**|| └**

**| └**

**└**

**scrie  $m$**

**2. Ce se afișează după rularea următoarei secvențe de instrucțiuni? (6p.)**

```
char s[50]="BAC informatica";
```

```
char *p,*q;
```

```
p= strtok(s, " ");
```

```
q= strtok(NULL, " ");
```

```
strcpy(s, strcat(q,p));
```

```
cout<<s;
```

3. Se consideră un graf neorientat cu 9 noduri, a cărui mulțime de muchii este  $M = \{(1,2), (1,9), (2,3), (3,4), (3,7), (3,8), (4,5), (5,6), (5,7), (6,7), (6,8), (8,9)\}$ . Indicați numărul minim de muchii și muchiile care pot fi eliminate astfel încât graful parțial obținut să fie eulerian și hamiltonian. (6p.)

**Subiectul al- III-lea****(30 de puncte)**

1. O bucată de pânză bicoloră este reprezentată sub forma unui tablou bidimensional cu  $n$  linii și  $n$  coloane ce conține numere naturale din mulțimea  $\{0,1\}$ . Se numește submatrice de coordonate  $(i_1, j_1, i_2, j_2)$  o zonă dreptunghiulară din matrice care are colțul stânga-sus pe linia  $i_1$ , coloana  $j_1$  și colțul dreapta-jos pe linia  $i_2$ , coloana  $j_2$  ( $i_1 \leq i_2, j_1 \leq j_2$ ). Dimensiunea unei submatrice este egală cu numărul de elemente din submatricea respectivă.

Spunem că pânza este perfectă dacă în matricea care o reprezintă există o submatrice de dimensiune maximă cu colțul stânga sus  $(i_1, j_1)$  și colțul dreapta jos  $(i_2, j_2)$  având toate elementele sale egale cu **0** și nu există elemente egale cu **0** în afara submatricei. Scrieți programul C/C++ care citește de la tastatură numărul natural  $n$  ( $2 \leq n \leq 23$ ) și apoi  $n \times n$  valori din mulțimea  $\{0,1\}$  reprezentând elementele tabloului bidimensional și afișează pe ecran mesajul **DA**, dacă pânza este perfectă sau mesajul **NU** în caz contrar. (10p)

**Exemplu:** pentru  $n=5$  și tabloul alăturat, se va afișa DA

```
1 1 1 1 1
```

```
1 0 0 0 1
```

```
1 0 0 0 1
```

```
1 1 1 1 1
```

```
1 1 1 1 1
```

**Exemplu:** pentru  $n=5$  și tabloul alăturat, se va afișa NU

1 1 1 1 1

1 0 0 0 1

1 0 0 0 1

1 1 1 1 1

1 0 0 1 1

2. Un număr natural nenul se numește **echilibrat** dacă suma cifrelor de pe poziții pare este egală cu suma cifrelor de pe poziții impare, cifrele se numerotează de la dreapta la stânga începând cu valoarea 0. Exemplu: **121** este număr echilibrat pentru că  $2=1+1$ . Scrieți în C/C++ definiția completă a subprogramului **echilibrat**, cu doi parametri, a și b, prin care primește câte un număr natural ( $2 \leq a < b \leq 10^9$ ,  $b-a \leq 10000$ ). Subprogramul afișează pe ecran, separate prin câte un spațiu, în ordine descrescătoare, toate numerele **echilibrate** din intervalul [a,b]. Dacă în interval nu există astfel de numere, subprogramul afișează pe ecran mesajul **nu exista**. (10p)

Exemplu: pentru  $a=100$   $b=150$ , se afișează pe ecran: **143 132 121 110**.

3. O secvență de **K** elemente a unui șir de numere naturale este numită secvență **RK**, dacă elementele din secvență dau resturi distincte la împărțirea cu **K**.

Fișierul **bac.txt** conține pe prima linie un număr natural **K** din intervalul [1,10], iar pe a doua linie conține un șir de cel puțin **K** și cel mult  $10^5$  numere naturale din intervalul  $[0,10^4]$ , separate prin câte un spațiu.

Se cere să se afișeze pe ecran suma maximă a elementelor unei secvențe **RK** din șirul de pe a doua linie a fișierului, dacă în șir există secvențe **RK** sau mesajul **NU EXISTA** dacă șirul din a doua linie a fișierului nu conține nicio secvență **RK**.

Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al spațiului de memorie utilizat.

**Exemplul 1.** Dacă fișierul conține numerele

3

10 10 11 3 4 2 49 4 2 3 21 27 12 13 atunci se afișează pe ecran 24

(secvența **RK 10 11 3** are suma maximă)

Exemplul 2. Dacă fișierul conține numerele

3

10 11 13 16 11 10 atunci se afișează pe ecran **NU EXISTĂ** (în șir nu există nicio secvență **RK**).

a) Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)

b) Scrieți programul C/C++ corespunzător algoritmului proiectat. (8p)

## Subiectul I

(20 de puncte)

1. d.  $!(z \geq a \ \&\& \ z \leq b) \ \&\& \ z \% 2 == 0$
2. b.  $p1.linia == p2.linia \ || \ p1.coloana == p2.coloana$
3. c. 4
4. c. 5
5. a. 12

## Subiectul II

(40 de puncte)

1. a) 2 (6p.)

b)  $n=5$   $a=50$   $b=100$  putem citi valorile 55 66 77 88 90 sau orice set format din 5 valori din care 4 sunt din mulțimea  $\{55, 66, 77, 88, 99\}$  (6p.)

c) `#include <iostream>``using namespace std;``int main()``{ int n, a, b, i, x, m=0;``cin >> n >> a >> b;``for(i=1; i<=n; i++)``{ cin >> x;``if(x >= a && x <= b)``if(x % 10 == x / 10) m++;``}``cout << m;``return 0;``}`

d) structura repetitivă cu test initial este structura cat timp ... executa (6p.)

citește  $n, a, b$  (numere naturale nenule) $m \leftarrow 0; i \leftarrow 1$ ┌ cat timp  $i \leq n$  execută| citește  $x$ | ┌ dacă  $x \geq a$  și  $x \leq b$  atunci|| ┌ dacă  $x \% 10 = [x / 10]$  atunci|| |  $m \leftarrow m + 1$ 

|| | └

|| └

|  $i \leftarrow i + 1$ 

└

scrie  $m$ 

2. informaticaBAC (6p.)

3. Numarul minim de muchii care pot fi eliminate este 3, muchiile care pot fi eliminate sunt  $[3,7], [3,8], [5,6]$  (6p.)

**Subiectul III****(30 de puncte)**

1. Vom determina în  $i1, j1$  coordonatele primului element 0 din matrice și în  $i2, j2$  coordonatele ultimului 0 din matrice, parcugem matrice și verificăm condițiile:

- elementele egale cu sunt în submatricea determinate de cele 2 puncte și
- elementele egale cu 1 sunt în afara submatricei.

```
#include <iostream>
using namespace std;
int main()
{
    int n, a[25][25], i, j, i1, j1, i2, j2, ok=1, primul=0;
    cin >> n;
    for(i=1; i<=n; i++)
        for(j=1; j<=n; j++)
            { cin >> a[i][j];
              if(a[i][j]==0 && primul==0)
                  { i1=i; j1=j; primul=1; }
              if(a[i][j]==0)
                  { i2=i; j2=j; }
            }
    if (primul == 0) { cout << "NU"; return 0; }
    for(i=1; i<=n && ok; i++)
        for(j=1; j<=n && ok; j++)
            if(a[i][j]==1)
                { if(i>=i1 && i<=i2 && j>=j1 && j<=j2)
                    ok=0;
                }
            else
                { if(!(i>=i1 && i<=i2 && j>=j1 && j<=j2))
                    ok=0;
                }
    if(ok && primul) cout << "DA";
    else cout << "NU";
    return 0;
}
```

2. Căutăm numerele echilibrate numai printre numerele divizibile cu 11.

```
void echilibrat(int a, int b){
    int ok=0, si, sp, ci, r;
    for(int i=b; i>=a; i--){
        if(i%11==0){
            ci=i; si=0; sp=0; r=0;
            while(ci){ if(r%2==0)
                sp=sp+ci%10;
                else
                si=si+ci%10;
                ci=ci/10; r++; }
            if(si==sp){ cout << i << ' '; ok=1; }
        }
    }
    if(!ok) cout << "nu exista";
}
```

3. a) O soluție eficientă utilizează un vector cu  $k$  elemente cu ajutorul căruia vom memora secvența curentă cu  $k$  elemente prin utilizarea vectorului circular, pentru a vedea dacă resturile la împărțirea cu  $k$  sunt distincte folosim un vector de apariții, dar și contorizăm câte resturi distincte avem în secvența curentă cu ajutorul variabilei  $nr$ , iar cu variabila  $s$  determinăm suma secvenței curente. La fiecare pas determinăm maximum acestor sume. Algoritmul este eficient din punct de vedere a spațiului de memorie deoarece utilizează doar variabile simple și 2 vectori a câte 10 elemente ( $k \leq 10$ ). Eficiența timp este dată de faptul că este un algoritm liniar, la o singură trecere prin fișier determină valoarea cerută, complexitatea algoritmului depinde doar de numărul de valori din fișier.

b)

```
#include <iostream>
#include<fstream>
using namespace std;
ifstream fin("bac.txt");
int k, i, x, v[10], s, smax,ap[10],nr,ok,y;
int main()
{  fin>>k;
   for(i=0;i<k;i++)
   {
       fin>>v[i];
       ap[v[i]%k]++;
       if(ap[v[i]%k]==1)nr++;
       s=s+v[i];
   }
   if(nr==k){smax=s;ok=1;}
   while(fin>>x)
   {
       y=v[i%k];s=s-y;
       if(ap[y%k]==1)nr--;
       ap[y%k]--;
       s=s+x;v[i%k]=x;ap[x%k]++;
       if(ap[x%k]==1)nr++;
       if(nr==k&& s>smax){smax=s;ok=1;}
       i++;
   }
   if(ok)cout<<smax;
   else cout<<"NU EXISTA";
   return 0;
}
```

**BAREM DE EVALUARE ȘI DE NOTARE****TEST 1**

- Se punctează oricare alte modalități de rezolvare corectă a cerințelor.
- Nu se acordă punctaje intermediare, altele decât cele precizate explicit prin barem. Nu se acordă fracțiuni de punct. Se acordă 10 puncte din oficiu. Nota finală se calculează prin împărțirea punctajului total acordat pentru lucrare la 10.
- Utilizarea unui tip de date care depășește domeniul de valori precizat în enunț este acceptată dacă acest lucru nu afectează corectitudinea în funcționarea programului.

**SUBIECTUL I****(20 de puncte)**

<b>1. d</b>	<b>2. b</b>	<b>3. c</b>	<b>4. c</b>	<b>5. a</b>	<b>5x4p.</b>
-------------	-------------	-------------	-------------	-------------	--------------

**SUBIECTUL al II - lea****(40 de puncte)**

<b>1.</b>	<b>a) Răspuns corect: 2</b>	<b>6p.</b>	
	<b>b) Pentru răspuns corect</b>	<b>6p.</b>	Orice set format din 5 valori din care 4 sunt din mulțimea {55, 66, 77, 88, 99}
	<b>c) Pentru program corect</b> -declaraire variabile -citire date -afișare date -instrucțiune de decizie -instrucțiuni repetitive -atribuiri -corectitudine globală a programului <sup>1)</sup>	<b>10p.</b> 1p. 1p. 1p. 2p. 3p. 1p. 1p.	
	<b>d) Pentru algoritm pseudocod corect</b> -echivalență a prelucrării realizate, conform cerinței (*) -corectitudine globală a algoritmului <sup>1)</sup>	<b>6p.</b> 5p. 1p.	(*) Se acordă numai 2p. dacă algoritmul are o structură repetitivă conform cerinței, principial corectă, dar nu este echivalent cu cel dat. Se va puncta orice formă corectă de structură repetitivă conform cerinței.
<b>2.</b>	<b>Pentru rezolvare corectă informaticaBAC</b>	<b>6p.</b>	(*) Se acordă câte 2p. pentru fiecare aspect (determinarea fiecărui cuvânt, ordinea cuvintelor)
<b>3.</b>	<b>Pentru rezolvare corectă</b> Numarul minim de muchii care pot fi eliminate este <b>3</b> , muchiile care pot fi eliminate sunt <b>[3,7],[3,8],[5,6]</b> (*)	<b>6p.</b>	(*) Se acordă câte 3 puncte pentru fiecare aspect al cerinței. Dacă doar două muchii sunt corecte, se acordă doar 2 puncte din cele 3

