

MODEL TEST 2

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.
- Identificatorii utilizați în rezolvări 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). Datele de intrare se consideră corecte, validarea lor nefiind necesară.

SUBIECTUL I

(20 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. Stabiliți care dintre expresiile următoare are valoarea 1 dacă și numai dacă variabila reală x aparține reuniunii intervalelor $[2,10]$ și $(30,40)$?

- a) $!((x < 2 \parallel x > 10) \parallel (x \geq 40 \parallel x \leq 30))$
 b) $(x \leq 10 \ \&\& \ x \geq 2) \ \&\& \ (x > 30 \ \&\& \ x < 40)$
 c) $(x > 10 \ \&\& \ x \geq 2) \ \&\& \ (x \leq 30 \ \&\& \ x \leq 40)$
 d) $!(x < 2 \parallel x > 10) \parallel !(x \geq 40 \parallel x \leq 30)$

2. Stabiliți ce valoare au variabilele x , y și z ce memorează numere naturale astfel încât expresia C++

$10/x/y+z\%5*x/3$ să aibă valoarea 200.

- a) $x=100, y=2, z=23$ b) $x=200, y=1, z=23$ c) $x=200, y=2, z=20$ d) $x=150, y=5, z=8$

3. Se consideră variabilele i, j și nr de tip întreg. Ce elemente va avea tabloul a , după execuția secvenței de program C/C++ următoare?

```
nr=0;
for(i=2;i<=3;i++)
for(j=2;j<=4;j++)
{
  if(i%j>j%i) a[nr]= i%j;
  else   a[nr]= j%i;
  nr++;
}
```

- a) (0,1,0,1,0,1) b) (0,2,0,2,0,2) c) (0,2,2,2,0,3) d) (1,1,2,1,1,1)

4. Se consideră tabloul unidimensional a , care conține, în ordine crescătoare, cele mai mici 5 numere prime de 2 cifre și tabloul unidimensional b care conține, în ordine crescătoare, cele mai mici 4 numere naturale de două cifre, care au suma cifrelor un număr prim. Prin interclasarea vectorilor a și b se va obține tabloul:

- a) (11,12,13,14,16,17,18,19,23) b) (11,11,12,13,14,15,19,19,23)
 c) (11,11,12,13,13,16,17,19,23) d) (11,11,12,13,14,16,17,19,23)

5. Se consideră algoritmul alăturat, descris în pseudocod.

S-a notat cu $x\%y$ restul împărțirii numerelor întregi x și y și cu $[x]$ partea întreagă a numărului real x .

Ce se va afișa prin execuția algoritmului, dacă $a=31287$ și $b=1887$?

- a. 187 b. 87 c. 3 d.4

citește a,b (a,b numere naturale nenule)

$y \leftarrow 0$

```

- repeta
  c1 ← a%10; c2 ← b%10
  dacă c1=c2 atunci
    y ← y+1
  a ← [a/10]
  b ← [b/10]
- până când a=0 sau b=0
scrie y

```

SUBIECTUL al II-lea
(40 puncte)
1. Se consideră algoritmul alăturat descris în pseudocod:

S-a notat cu $x \% y$ restul împărțirii numerelor întregi x și y și cu $[x]$ partea întreagă a numărului real x .

a) Ce se va afișa, dacă se citesc, în această ordine, numerele **5,3,2324,31,7229,361,75** (6p.).

b) Scrieți un set de date de intrare care să determine afișarea valorii **200**.

(6p.)

c) Scrieți programul C/C++ corespunzător algoritmului dat.

(10p.)

d) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să se înlocuiască structura **pentru** cu alt tip de structură repetitivă.

(6p.)

citește n,c (n, c numere naturale nenule, $c < 10$)

$e \leftarrow 0$

```

- pentru i ← 1, n execută
  citește x (numar natural nenul)
  p ← 1
  nr ← 0
  cx ← x
  cat timp x ≠ 0 executa
    dacă x%10 ≤ c atunci
      nr ← nr + (x%10)*p
      p ← p*10
    x ← [x/10]
  dacă nr%2=0 și nr > 0 atunci
    e ← e + cx
scrie e

```

2. Se consideră declarațiile de mai jos, în care variabilele **numarator1** și **numitor1** memorează două numere naturale nenule ce reprezintă numărătorul și numitorul primei fracții iar **numarator2** și **numitor2** memorează numărătorul și numitorul celei de a doua fracții.

int numarator1,numitor1,numarator1,numitor2,numarator_suma,numitor_suma;

Scrieți o secvență de instrucțiuni C/C++ care determină suma celor două fracții, memorând în variabilele **numarator_suma** și **numitor_suma** numărătorul și numitorul sumei celor două fracții. (6p.)

3. În secvența de instrucțiuni de mai jos, variabilele **i** și **j** sunt de tip întreg. Fără a utiliza alte variabile, scrieți una sau mai multe instrucțiuni care pot înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, să se afișeze caracterele de mai jos, în această ordine. (6p.)

```
for (i=0; i<5; i++) {
    for (j=0; j<5; j++)
        .....
}
```

0	1	2	3	4
1	2	3	4	0
2	3	4	0	1
3	4	0	1	2
4	0	1	2	3

SUBIECTUL al III - lea**(30 puncte)**

1. Scrieți un program C/C++ care citește de la tastatură un număr natural **n** ($1 \leq n \leq 100$) și un tablou unidimensional **v**, format din **n** numere naturale nenule și distincte, fiecare număr având cel mult 9 cifre și afișează numărul de perechi distincte formate din elemente din tabloul **v**, care sunt prime între ele. Două numere naturale sunt prime între ele dacă au un singur divizor comun: numărul 1. Două perechi de elemente sunt distincte dacă diferă prin cel puțin un element. **Exemplu:** dacă **n=6** și **v=(4,22,12,9,63,28)** atunci se va afișa valoarea **5**, deoarece vectorul are 5 perechi de numere ce verifică proprietatea dată: **(4,9);(4,63);(22,9);(22,63);(9,28)**. (10p.)

2. Se citește de la tastatură un număr natural **n** de cel mult 9 cifre, care are cel puțin o cifră impară. Scrieți un program în limbaj pseudocod care determină numărul **m** obținut prin eliminarea cifrelor pare ale numărului **n** și afișează oglinditul numărului **m**. **Exemplu:** dacă **n=5768132**, se va obține **m=5713** și se va afișa oglinditul său egal cu **3175**. (10p.)

3. Fișierul BAC.TXT conține pe prima linie două numere naturale nenule **n** și **m** ($n \leq 1000$, $m \leq 10000$). A doua linie din fișier conține un șir **a** având **n** numere naturale, formate din cel mult trei cifre, în ordine strict crescătoare și separate prin câte un spațiu. A treia linie din fișier conține un șir **b** având **m** numere naturale, formate din cel mult trei cifre și separate prin câte un spațiu. Scrieți un program C/C++ care citește valorile din fișierul BAC.TXT și afișează pe ecran, în ordine crescătoare și separate prin spațiu, numerele din șirul **b** ce ar putea fi inserate în șirul **a**, astfel încât **a** să rămână ordonat strict crescător sau afișează mesajul **Nu exista numere care vor fi inserate**, dacă niciun număr din șirul **b** nu poate fi inserat în șirul **a** cu proprietatea de mai sus. Se va utiliza un algoritm eficient din punct de vedere al memoriei utilizate și al timpului de execuție. De exemplu, dacă fișierul BAC.TXT are conținutul următor, pe ecran se va afișa **3 35 90**.

6 7
5 14 25 80 100 150
5 3 90 14 35 3 35

a. Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia.

(2p.)b. Scrieți programul C/C++ corespunzător algoritmului descris la punctul a.

(8p.)

REZOLVARE SUBIECTE
TEST 2
Subiectul I

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

Subiectul al II-lea

1.

a) 9553

b) Algoritmul adună dintre cele n numerele citite pe cele care respectă condiția: numărul obținut prin eliminarea cifrelor strict mai mari decât c este par. Un set de numere posibile: 5, 6, 34, 128, 12, 26, 179, deoarece se va obține suma egală cu $34+128+12+26=200$

c) Programul C++:

```
#include <iostream>
using namespace std;
int main(){
    unsigned int n,c,e,i,x,p,nr,cx;
    cin>>n>>c;
    e=0;
    for(i=1;i<=n;i++){
        cin>>x;
        p=1;
        nr=0;
        cx=x;
        while(x!=0){
            if(x%10<=c){
                nr=nr+(x%10)*p;
                p=p*10;
            }
            x=x/10;
        }
        if(nr%2==0 && nr>0)
            e=e+cx;
    }
    cout<<e;
    return 0;
}
```

d) Programul în limbaj pseudocod echivalent este:

```
citește n,c (n, c numere naturale nenule, c<10)
e←0
i←1
cât timp i≤n execută
    p←1
    nr←0
    cx←x
    cât timp x≠0 execută
        dacă x%10≤c atunci
            nr←nr+(x%10)*p
            p←p*10
        x←[x/10]
    dacă nr%2=0 și nr>0 atunci
        e←e+cx
    i←i+1
scrie e
```

2. numarator_suma=numarator1*numitor2+numarator2*numitor1;
numitor_suma=numitor1*numitor2;
3. cout<<(i+j)%5<<' '; cout<<endl;

Subiectul al III-lea

1.

```

#include<iostream>
using namespace std;
int main(){
    int a,b,r,i,j,v[101],n,nr;
    cout<<"n=";
    cin>>n;
    for(i=1;i<=n;i++){
        cout<<"v["<<i<<"]="";
        cin>>v[i];
    }
    nr=0;
    for(i=1;i<n;i++){
        for(j=i+1;j<=n;j++){
            a=v[i];
            b=v[j];
            while(b!=0){
                r=a%b;
                a=b;
                b=r;
            }
            if(a==1)
                nr++;
        }
    }
    cout<<"Numarul de perechi este: "<<nr;
    return 0;
}

```

2. citește n (n număr natural)

 $m \leftarrow 0$ $p \leftarrow 1$

cât timp (n ≠ 0) execută

dacă (n%2=1) atunci

 $m \leftarrow n \% 10 * p + m$ $p \leftarrow p * 10$ $n \leftarrow [n/10]$ $r \leftarrow 0$

cât timp (m ≠ 0) execută

 $r \leftarrow r * 10 + m \% 10$ $m \leftarrow [m/10]$

scrie r

3. a. Programul utilizează un vector v cu 1000 de elemente, corespunzător valorilor posibile ale numerelor citite, în care se completează valoarea 1 pentru numerele din șirul a simultan cu citirea lor. La citirea fiecărui număr x din șirul b se verifică dacă $v[x]=0$ (ce semnifică faptul că nu există valoarea x în șirul a și poate fi inserată) și se completează cu valoarea 2. Algoritmul este eficient ca spațiu de memorie deoarece utilizează un singur vector și este eficient ca timp de execuție deoarece parcurge o singură dată fiecare șir de numere (algoritm liniar).

b.

```
#include<iostream>
#include<fstream>
using namespace std;
int v[1000];
int main(){
    ifstream fin("BAC.TXT");
    int n,m,i,a,b,ok;
    ok=0;
    fin>>n>>m;
    for(i=1;i<=n;i++){
        fin>>a;
        v[a]=1;
    }
    for(i=1;i<=m;i++){
        fin>>b;
        if(v[b]==0){
            v[b]=2;
            ok=1;
        }
    }
    if(ok==0)
        cout<<"Nu exista numere care vor fi inserate";
    else{
        for(i=0;i<=999;i++)
            if(v[i]==2)
                cout<<i<<' ';
    }
    return 0;}

```

BAREM DE EVALUARE ȘI DE NOTARE**TEST 2**

- 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)**

1. d	2. b	3. c	4. d	5. c	5x4p.
------	------	------	------	------	-------

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

1.	a) Răspuns corect: 9553	6p.	
	b) Pentru răspuns corect	6p.	
	c) Pentru program corect -declarare variabile -citire date -afișare date -instrucțiune de decizie -instrucțiuni repetitive (*) -atribuiri -corectitudine globală a programului ²⁾	10p. 1p. 1p. 1p. 2p. 3p. 1p. 1p.	(* Se acordă numai 2p. dacă numai una dintre instrucțiunile repetitive este conform cerinței.
	d) Pentru algoritm pseudocod corect -echivalență a prelucrării realizate, conform cerinței (*) -corectitudine globală a algoritmului ²⁾	6p. 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.
2.	Pentru rezolvare corectă -determinarea numitorului sumei fracțiilor -determinarea numărătorului sumei fracțiilor -corectitudine globală a secvenței ¹⁾	6p. 3p. 2p. 1p.	
3.	Pentru rezolvare corectă -atribuire a valorilor indicate pentru afișare (*) -afișarea valorilor pe linii, conform cerinței	6p. 4p. 2p.	(* Se acordă numai 2p. dacă doar o parte dintre elementele afișate sunt conform cerinței.

¹ Corectitudinea globală vizează structura, sintaxa și alte greșeli neprecizate în barem.

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

<p>Pentru program corect -declararea corectă a variabilei de tip tablou</p> <p>1. -citire corectă a datelor de intrare - algoritm de verificare dacă două numere sunt prime între ele - determinarea numărului de perechi cerut -corectitudine globală a programului ²⁾</p>	<p>10p. 1p. 1p. 4p. 3p. 1p.</p>	
<p>Pentru program corect -declararea de variabile simple -citirea datelor -determinarea numărului m obținut după eliminarea cifrelor pare -accesul la cifrele numărului n -determinarea oglinditului numărului m -corectitudine a globală a programului ²⁾</p>	<p>10p. 1p. 1p. 2p. 2p. 3p. 1p.</p>	
<p>a) Pentru răspuns corect -coerența descrierii algoritmului (*) -justificare a unor elemente de eficiență</p> <hr/> <p>b) Pentru program corect -operații cu fișiere: declarare, pregătire în vederea citirii, citire din fișier -determinare a valorii cerute (*),(**) 3. -utilizarea unui algoritm eficient (***) -declararea variabilelor, afișarea datelor, corectitudinea globală a programului ²⁾</p>	<p>2p. 1p. 1p.</p> <p>8p. 1p. 5p. 1p. 1p.</p>	<p>.(*) Se acordă punctajul chiar dacă algoritmul ales nu este eficient.</p> <p>(**) Se acordă numai 3p. dacă algoritmul este principial corect, dar nu oferă rezultatul cerut pentru toate seturile de date de intrare.</p> <p>(***) Se acordă punctajul numai pentru un algoritm liniar care utilizează eficient memoria. O soluție posibilă construiește, pe măsura citirii datelor, un vector caracteristic pentru elementele din șirul a și le notează cu 1. Apoi citește elementele șirului b și verifică pentru fiecare număr, pe baza vectorului caracteristic, dacă nu există un termen egal cu el din șirul a, în caz afirmativ acesta se notează cu 2 în vector. Se parcurge vectorul caracteristic, în sens crescător și se afișează numerele notate cu 2.</p>