

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.
- În grafurile din cerințe oricare arc/muchie are extremități distincte și oricare două arce/muchii diferă prin cel puțin una dintre extremități.

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

2. Ce valoare are apelul $\text{sir}(s,0)$ al funcției următoare, dacă șirul de caractere s este **a1de2d33bc**?

```
int sir(char s[],int i){
  if (i<strlen(s)-1)
    if (s[i]>='a' && s[i]<='z' && s[i+1]<='9' && s[i+1]>='0')
      return 1+sir(s,i+1);
    else
      return sir(s,i+1);
  else return 0;
}
```

- a. 6 b. 5 c. 3 d. 4

3. Un graf neorientat are 7 muchii și conține două componente conexe. Fiecare componentă conexă din graf este arbore. Câte noduri are acest graf?

- a. 8 b. 10 c. 9 d. 11

4. Utilizând metoda backtracking, se generează, toate modalitățile de scriere a unui număr natural mai mare decât 1, ca sumă de numere prime, în ordine crescătoare. Pentru numărul 7, se generează în ordine combinațiile: 2+2+3, 2+5, 7. Utilizând același algoritm se generează toate modalitățile de scriere a numărului 18 ca sumă de numere prime, în ordine crescătoare. Care este a șasea soluție generată?

- a. 2+2+2+2+5+5 b. 2+2+2+3+3+3+3 c. 2+2+7+7 d. 2+2+2+5+7

5. Un graf neorientat are 7 noduri și este reprezentat prin listele de adiacență alăturate. Care dintre afirmațiile următoare este adevărată pentru acest graf?

- Graful este hamiltonian și eulerian
- Există un subgraf hamiltonian și eulerian format din 4 noduri al grafului dat
- Graful este complet
- Există un subgraf complet format din 4 noduri al grafului dat

1: 2,3,4,5

2: 1,3,4

3: 1,2,5

4: 1,2,5

5: 1,3,4

SUBIECTUL a II-lea**(40 de 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 $i \leftarrow 1, n$ execută

citește x (număr natural nenul)

 $p \leftarrow 1$ $nr \leftarrow 0$ $cx \leftarrow x$ cat timp $x \neq 0$ executadaca $x \% 10 \leq c$ atunci $nr \leftarrow nr + (x \% 10) * p$ $p \leftarrow p * 10$ $x \leftarrow [x/10]$ daca $nr \% 2 = 0$ și $nr > 0$ atunci $e \leftarrow e + cx$ scrie e

2. Se consideră declarațiile de mai jos, în care tabloul unidimensional c memorează informații despre 200 de concurenți participanți la un concurs sportiv. Pentru fiecare concurent se memorează: numele prin câmpul **nume_concurent**, denumirea probei la care participă prin câmpul **proba** și punctajul obținut prin câmpul **punctaj**. Denumirea unei probe poate conține mai multe cuvinte separate prin spațiu. Variabila p memorează denumirea unei probe din concurs.

```
struct concurs{
    char nume_concurent[25],proba[50];
    int punctaj;
} c[200];
char p[50];
```

Scrieți o secvență de instrucțiuni C/C++ care citește variabila p , determină și afișează pe ecran, separate prin virgulă, numele concurenților care au obținut un punctaj egal cu 100 la proba cu denumirea p . (6p.)

3. Variabilele i și j sunt de tip întreg, iar variabila a memorează un tablou bidimensional cu 5 linii și 5 coloane, numerotate de la 0 la 4. Fără a utiliza alte variabile decât cele menționate, scrieți în limbajul C/C++, secvența de instrucțiuni de mai jos, înlocuind punctele de suspensie astfel încât, în urma execuției secvenței obținute, variabila a să memoreze tabloul alăturat.

```
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 de puncte)**

1. Subprogramul **perechi** primește prin parametrul n un număr natural ($2 \leq n \leq 100$), prin parametrul v primește un tablou unidimensional, format din n numere naturale nenule și distincte, fiecare număr având cel mult 9 cifre și prin parametrul p furnizează 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. Scrieți în limbajul C/C++ definiția completă a subprogramului **perechi**.

Exemplu: dacă $n=6$ și $v=(4,22,12,9,63,28)$ atunci, după apel, se obține $p=5$, deoarece vectorul are 5 perechi de numere ce verifică proprietatea dată: $(4,9);(4,63);(22,9);(22,63);(9,28)$. **(10p.)**

2. Se consideră un număr natural nenul $n(n \leq 30)$, un cuvânt c , format din cel mult 20 de litere mici și un text format din n linii. Fiecare linie din text are cel mult 200 de caractere care sunt: litere din alfabetul englez, spații și se încheie cu caracterul newline. Fiecare linie din text este formată din mai multe cuvinte, separate între ele printr-un spațiu. Datele se citesc de la tastatură astfel: de pe prima linie se citește numărul n și cuvântul c , separate printr-un spațiu, de pe fiecare dintre următoarele n linii se citește câte o linie din text. Scrieți un program C/C++ care citește datele menționate, determină și afișează pe ecran pentru fiecare linie din text numărul de cuvinte ce au ca prefix cuvântul c , fără a face diferență între litere mari și litere mici. Cele n numere naturale afișate, corespunzător celor n linii din text, vor fi separate printr-un spațiu. De exemplu, dacă n este 4, cuvântul c este **cod** și textul este

Se poate cauta **codul** postal al unei locuințe și **codurile** postale ale locuințelor unei strazi
 Ai **codificat** și decodificat corect textul dat
 Adriana a recitit **Codul** Muncii **Codul** fiscal și **Codul** Administrativ
Codrin a mers la medic unde a primit un **cod** de diagnostic și apoi a recodat un modul din monocrom

programul va afișa **2 1 3 2**, deoarece prima linie are 2 cuvinte ce au ca prefix cuvântul **cod**, a doua linie are un cuvânt ce are ca prefix cuvântul **cod**, a treia linie are 3 cuvinte ce au ca prefix cuvântul **cod** și a patra linie are 2 cuvinte ce are ca prefix cuvântul **cod**. **(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.)**

SUBIECTUL I

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

SUBIECTUL al II-lea

1. Rezolvare:

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. cin.getline(p,50);
for(i=0;i<200;i++){
    if(strcmp(p,c[i].proba)==0 && c[i].punctaj==100){
        cout<<c[i].nume_concurent<<',';
    }
}
```

3. $a[i][j]=(i+j)\%5;$

SUBIECTUL al III-lea

1.

```

void perechi(int n, int v[], int &p){
    int a,b,r,i,j;
    p=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)
                p++;
        }
    }
}

```

2.

```

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

int main()
{
    char c[21], s[201],*p;
    int n,nr,i,j;
    cin>>n>>c;
    cin.get();
    for(i=1; i<=n; i++)
    {
        cin.getline(s,201);
        for(j=0; j<strlen(s); j++)

            if(s[j]>='A' && s[j]<='Z')
            {
                s[j]=s[j]+32;
            }
        nr=0;
        p=strtok(s," ");
        while(p!=NULL)
        {
            if(strstr(p,c)==p)
                nr++;
            p=strtok(NULL," ");
        }
        cout<<nr<<' ';
    }
    return 0;
}

```

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 $v[x]$ 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)**

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

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

	a) Răspuns corect: 9553	6p.	
	b) Pentru răspuns corect	6p.	
1.	c) Pentru program corect -declaraire variabile -citire date -afișare date -instrucțiune de decizie -instrucțiuni repetitive (*) -atribuiri -corectitudine globală a programului ¹⁾	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 ¹⁾	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 -citire corectă a denumirii probei p -acces la câmpurile înregistrării -afișarea rezultatelor cerute (*) -corectitudine globală a secvenței de program	6p. 1p. 1p. 3p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect (determinarea numelor concurenților, determinarea probei p, determinarea punctajului egal cu 100,)
3.	Pentru rezolvare corectă -acces la un element al tabloului -atribuire a valorilor indicate elementelor tabloului (*)	6p. 1p. 5p.	(*) Se acordă numai 2p. dacă doar o parte dintre elementele tabloului sunt conform cerinței.

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

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

1.	Pentru subprogram corect -antet subprogram (*) -determinare a numărului cerut (**) -returnarea rezultatului prin parametru -declarare a variabilelor locale, -corectitudine globală a subprogramului ¹⁾	10p. 2p. 6p. 1p. 1p.	.(*) Se acordă câte 1p. pentru fiecare aspect al antetului (structură, parametri) conform cerinței. (**) Se acordă câte 3p. pentru fiecare aspect al cerinței (algoritm de verificare dacă două numere sunt prime între ele, algoritm de determinare a valorii p).
2.	Pentru program corect -declararea variabilelor -citire a datelor -determinare a șirului de valori cerut (*) -declarare a variabilelor simple, -corectitudine a globală a programului ¹⁾	10p. 1p. 2p. 6p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect al cerinței (acces la o linie din text, cuvinte suport, identificare prefix, determinare număr de cuvinte care sunt prefixe, afișarea numărului de prefixe din linie, utilizarea tuturor liniilor din text).
3.	a) Pentru răspuns corect -coerența descrierii algoritmului (*) -justificare a unor elemente de eficiență	2p. 1p. 1p.	.(*) Se acordă punctajul chiar dacă algoritmul ales nu este eficient. (**) Se acordă numai 3p. dacă algoritmul este
	b) Pentru program corect -operații cu fișiere: declarare, pregătire în vederea citirii, citire din fișier -determinare a valorii cerute (*),(**) -utilizarea unui algoritm eficient (***) -declarare a variabilelor, afișare a datelor, corectitudine globală a programului ¹⁾	8p. 1p. 5p. 1p. 1p.	principial corect, dar nu oferă rezultatul cerut pentru toate seturile de date de intrare. (***) 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.