Teza 1

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 reale, iar **a ≤ b.** Care dintre expresiile următoare C/C++ are valoarea **1** dacă și numai dacă valoarea variabilei **x nu** aparține intervalului închis determinat de valorile variabilelor **a** și **b**?
   a) `(x>a) || (x>b)` **b)** `(x<a) || (x>b)` **c)** `x<a && x>b` **d)** `x>=a && x<=b`

2. Numim înălțime a unui arbore cu rădăcină numărul de muchii ale celui mai lung lanț elementar care are una dintre extremități în rădăcina arborelui. Înălțimea arborelui cu rădăcină, având **8** noduri, numerotate de la **1** la **8**, reprezentat prin vectorul „de tați" **(6, 6, 5, 0, 6, 4, 4, 7)** este:
   a) 2 **b)** 3 **c)** 4 **d)** 5

3. Se consideră algoritmul care determină toate permutările distincte de n obiecte (numerotate de la 1 la n) în care nu există puncte fixe. O permutare (p1, p2, ..., pn) are puncte fixe dacă există cel puțin o componentă pi = i. De exemplu, pentru n = 5, permutarea (2, 3, 5, 4, 1) are puncte fixe deoarece p4 = 4. Pentru n = 4, stabiliți câte permutări fără puncte fixe există.
   a) 8 **b)** 12 **c)** 10 **d)** 9

4. Se consideră un graf neorientat cu 20 noduri cu proprietatea că gradul fiecărui nod este mai mare sau egal cu 3. Care este numărul maxim de componente conexe pe care le poate avea acest graf?
   a) 4 **b)** 6 **c)** 1 **d)** 5

5. Se consideră subprogramul f definit de mai jos.

```
void f (int n)
  { if (n!=0)
    {if (n%10>4)cout<<n%10;
     f (n/10) ;
     cout<<n%10;
    }
  }
```

   Ce se va afișa în urma apelului f(54621)?
   a) 6554621 **b)** 654621 **c)** 6655421 **d)** 1266455

SUBIECTUL II (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.

```
citește a, n
(a număr întreg, n număr natural)
p←1
cât timp n>0 execută
    dacă n%2=0 atunci
        a←a*a
        n←[n/2]
    altfel
        p←p*a
        n←n-1
scrie p
```

a) Scrieți valoarea care se va afișa pentru a = 2 și n = 7. **(6 puncte)**

b) Scrieți două seturi distincte de valori pentru a și n astfel încât rezultatul afișat să fie 1. **(6 puncte)**

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

d) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să se utilizeze alt tip de structură repetitivă. **(6 puncte)**

2. Se consideră declarările de mai jos, în care variabila **e** memorează numele și data nașterii unui elev. Scrieți o secvență de instrucțiuni **C/C++** care citește de la tastatură informațiile despre un elev în variabila **e** și afișează numele elevului dacă anul nașterii este egal cu **2000** sau data nașterii în caz contrar, informațiile din dată se afișează pe aceeași linie separate printr-un spațiu. **(6 puncte)**

```
struct data
{int zi,luna, an;};
struct elev
{ char nume[21];
  data dn;
}
elev e;
```

3. În secvența de instrucțiuni de mai jos variabilele i și j sunt de tip întreg, iar variabila A memorează un tablou bidimensional cu 5 linii și 5 coloane, numerotate de la 1 la 5. Elementele tabloului sunt numere întregi. Fără a utiliza alte variabile, scrieți una sau mai multe instrucțiuni care pot înlocui punctele de suspensie astfel încât în urma executării secvenței obținute, tabloul memorat în variabila A să aibă elementele din figura de mai jos. **(6 puncte)**

```
for(i=1;i<=5;i++)
  for(j=1;j<=5;j++)
     ...............
```

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

SUBIECTUL III (30 de puncte)

1. Scrieți un program C/C++ care citește de la tastatură un șir de caractere format din cel mult 100 de caractere (litere mici și mari ale alfabetului englez și caracterul #) și afișează pe ecran șirul obținut prin inversarea ordinii literelor tuturor cuvintelor de lungime maximă. Șirul începe și se termină cu caracterul #, ca în exemplu. Un cuvânt din șir reprezintă o succesiune de litere delimitate de două caractere #.
   Exemplu: dacă se citește șirul de caractere **#Voi#da#bacu#la#info#** se va afișa
   #Voi#da#ucab#la#ofni# **(10 puncte)**

2. Subprogramul **numărare** are patru parametri:
   - **n**, prin care primește un număr natural ($2 \le n \le 20$);
   - **m**, prin care primește un număr natural ($2 \le m \le 20$);
   - **a**, prin care primește un tablou unidimensional care memorează un șir de **n** numere naturale, fiecare cu cel mult **4** cifre;
   - **b**, prin care primește un tablou unidimensional care memorează un șir de **m** numere naturale, fiecare cu cel mult **4** cifre.

   Subprogramul returnează numărul de elemente din tabloul a, care sunt strict mai mici decât toate elementele din tabloul b. Scrieți în limbajul **C/C++** definiția completă a subprogramului **numărare**.
   Exemplu: dacă **n = 7**, **m = 8** și **a = (1, 4, 5, 3, 82, 6, 2)**, **b = (56, 6, 34, 23, 8, 9, 12, 18)** atunci, după apel, subprogramul va returna valoarea **5** (valorile **1, 4, 5, 3, 2** din tabloul a sunt strict mai mici decât toate valorile din tabloul b. **(10 puncte)**

3. Numim **secvență fazan** a unui șir de numere naturale un subșir al acestuia, format din termeni aflați pe poziții consecutive în șirul dat cu proprietatea că prima cifră a termenului curent este egală cu ultima cifră a termenului anterior. Lungimea secvenței este egală cu numărul de termeni ai acesteia. Fișierul **`bac.txt`** conține un șir de cel puțin două și cel mult 10^9 de numere naturale din intervalul $[0, 10^9]$. Numerele sunt separate prin câte un spațiu, iar în șir există cel puțin doi termeni fazan pe poziții consecutive. Se cere să se determine o **secvență fazan** de lungime maximă în șirul aflat în fișier și să se afișeze pe ecran lungimea acestei secvențe. Pentru determinarea secvenței cerute se utilizează un algoritm eficient din punctul de vedere al memoriei necesare și al timpului de executare.
   Exemplu: dacă fișierul **`bac.txt`** conține numerele 12 **13 31 123 321** 61 76 **25 54 425 511 121** 311 311 atunci pe ecran se afișează valoarea **5.**
   a) Se cere să se proiecteze un algoritm eficient din punct de vedere al timpului de executare și al spațiului de memorie utilizat și să se realizeze o descriere de 3-4 rânduri a algoritmului ales justificându-se eficiența acestuia. **(2 puncte)**
   b) Scrieți programul C/C++ corespunzător algoritmului descris. **(8 puncte)**

Teza 2

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 reale. Care dintre următoarele instrucțiuni **C/C++** atribuie variabilei z valoarea maximă dintre valorile variabilelor a și b?
   a) `z=a>b?a:b;`
   b) `z=a<b?a:b;`
   c) `z=(a+b)/2-(a-b)/2`
   d) `z=(a+b)/2;`

2. Într-un graf neorientat cu 10 noduri, fiecare nod are gradul 2. Care este numărul maxim de componente conexe din care poate fi format graful?
   a) 6 **b)** 3 **c)** 5 **d)** 1

3. Se generează toate submulțimile cu 2 elemente ale mulțimii {1, 2, 3, 4} în ordinea 1 2, 1 3, 1 4, 2 3, 2 4, 3 4. Dacă se utilizează exact aceeași metodă pentru a genera submulțimile de trei elemente ale mulțimii {5, 6, 7, 8}, atunci antepenultima submulțime este:
   a) 5 7 8 **b)** 5 6 8 **c)** 6 7 8 **d)** 5 6 7

4. Care este numărul maxim de arce într-un graf orientat cu 10 noduri?
   a) 100 **b)** 20 **c)** 45 **d)** 90

5. Se consideră subprogramul **f** definit astfel:

```
void f (int n, int &m)
{if (n!=0)
        {if (n%10>m)m=n%10;
        f(n/10,m) ;
        cout<<m<<' ';
        }
}
```

Ce se va afișa pentru m = 0 în urma apelului f(54321, m)?
a) 5 4 3 2 1 **b)** 1 2 3 4 5 **c)** 5 5 5 5 5 **d)** 5 4 3 1 2

SUBIECTUL II (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.

```
citește n (n număr natural)
y←0
cât timp n>0 execută
  citește x
  cât timp x%2 = 0 execută
    x←x/2
  dacă x=1 atunci
    y←y+1
  n←n-1
scrie y
```

a) Scrieți valoarea care se va afișa dacă se citesc pe rând valorile 4, 8, 6, 16, 45. **(6 puncte)**

b) Pentru n = 4, scrieți un set de date de astfel încât rezultatul afișat să fie 0. **(6 puncte)**

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

d) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să se utilizeze alt tip de structură repetitivă. **(6 puncte)**

2. Se consideră declarările de mai jos, în care variabila **p** memorează numărul de vârfuri și coordonatele carteziene ale vârfurilor unui poligon convex, variabila i este de tip int. Scrieți o secvență de instrucțiuni **C/C++** care înlocuind punctele de suspensie citește de la tastatură coordonatele vârfurilor poligonului. **(6 puncte)**

```
struct punct
{int x,int y;};
struct poligon
{ int nr_vf ;
  punct vf[20];}p;

cin>>p.nr_vf;
for(i=0;i<p.nr_vf;i++)
    ......................
```

3. În secvența de instrucțiuni de mai jos variabilele i și j sunt de tip întreg, iar variabila A memorează un tablou bidimensional cu 5 linii și 5 coloane, numerotate de la 1 la 5. Elementele tabloului sunt numere întregi. Fără a utiliza alte variabile, scrieți una sau mai multe instrucțiuni care pot înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, tabloul memorat în variabila A să aibă elementele din figura de mai jos. **(6 puncte)**

```
for(i=1;i<=5;i++)
  for(j=1;j<=5;j++)
    ............
```

```
1 1 1 1 3
1 1 1 3 2
1 1 3 2 2
1 3 2 2 2
3 2 2 2 2
```

SUBIECTUL III (30 de puncte)

1. Scrieţi un program C/C++ care citeşte de la tastatură un şir de caractere format din cel mult 100 de caractere (litere mici, litere mari ale alfabetului englez şi spaţii) şi afişează pe ecran litera care apare de cele mai multe ori şi numărul ei de apariţii, ca în exemplu. Dacă sunt mai multe litere care apar de acelaşi număr maxim de ori se vor afişa toate.
Exemplu: Dacă se citeşte şirul de caractere **`Voi da Bacalaureatul la Informatica`** se va afişa **a 8**. **(10 puncte)**

2. Subprogramul **`numărare`** are doi parametri:
   - a, prin care primeşte un număr natural ($2 \le a \le 20000$);
   - b, prin care primeşte un număr natural ($2 \le b \le 20000$);

   Subprogramul returnează numărul de valori din intervalul închis determinat de a şi b, care au exact 3 divizori. Scrieţi în limbajul C/C++ definiţia completă a subprogramului **`numărare`**.
   Exemplu: Dacă a = 6 şi b = 36, subprogramul va returna 2. **(6 puncte)**

3. Din fişierul **`bac.txt`** se citesc `n` şi `m` (numere naturale, `0<m<n<1000000`) de pe prima linie, apoi `n` numere naturale cu cel mult două cifre fiecare `a1, a2, ..., an` de pe linia a doua şi apoi `m` numere naturale cu cel mult două cifre fiecare `b1, b2, ..., bm` de pe linia a treia a fişierului. Să se determine câte şiruri `b` se pot obţine din şirul `a` dacă se poate schimba ordinea elementelor din şirul `a`. Se va afişa pe ecran numărul numărul de şiruri obţinute.
   a) Se cere să se proiecteze un algoritm eficient din punct de vedere al timpului de executare şi să se realizeze o descriere de 3-4 rânduri a algoritmului ales justificându-se eficienţa acestuia. **(2 puncte)**
   b) Să se scrie programul C/C++ ce realizează prelucrarea descrisă şi afişează pe ecran un mesaj corespunzător. **(8 puncte)**
   De exemplu, pentru fişierul **`bac.txt`** cu conţinutul:
   8 3
   1 6 3 1 3 7 6 1
   6 1 3
   se afişează valoarea **2**.

Teza 3

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** sunt numere naturale. Care dintre expresiile **C/C++** are valoarea 1 dacă și numai dacă valorile variabilelor a și b sunt numere naturale consecutive?

a) `a-b==1`

b) `a==1&&b==0`

c) `a-b==1&&b-a==1`

d) `a-b==1||b-a==1`

2. Se consideră un arbore cu rădăcină, în care rădăcina se află pe nivelul 0 și orice nod de pe nivelul i are exact i + 1 descendenți, cu excepția nodurilor de pe nivelul 3 care sunt noduri terminale. Numărul de noduri frunze ale arborelui sunt:

a) 5 **b)** 10 **c)** 6 **d)** 7

3. Se utilizează metoda backtracking pentru a genera toate submulțimile mulțimii {1, 2, 3, 4, 5}. Câte submulțimi care conțin elementul 2 și nu conțin elementul 4 sunt generate?

a) 8 **b)** 12 **c)** 10 **d)** 16

4. Într-un graf neorientat cu 10 noduri, există muchii între nodurile `i, j` care au proprietatea că `abs(i-j)>0`. Numărul de valori egale cu 1 din matricea de adiacență este:

a) 91 **b)** 100 **c)** 90 **d)** 80

5. Se consideră subprogramul f definit astfel:

```
void f (int n)
  { if (n!=0)
      {
          f(n/2);
          cout<<n%2;
      }
  }
```

Ce se va afișa în urma apelului f(43)?

a) 11011 **b)** 110101 **c)** 1101011 **d)** 101011

SUBIECTUL II (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.

```
citește x (x număr natural)
nr←0
cât timp x>0 execută
    dacă x%2=0 atunci
        nr←nr+1
    x←[x/2]
scrie nr
```

a) Scrieți valoarea care se va afișa pentru x =13 **(6 puncte)**

b) Scrieți două valori distincte x astfel încât rezultatul afișat să fie 0. **(6 puncte)**

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

d) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să se utilizeze alt tip de structură repetitivă. **(6 puncte)**

2. Se consideră declarările de mai jos, în care variabilele **E1** și **E2** memorează numele și data nașterii a doi elevi. Scrieți o secvență de instrucțiuni **C/C++** care afișează numele elevului mai mare, știind că ambii elevi sunt născuți în același an, în zile diferite. **(6 puncte)**

```
struct data
{int zi, luna, an;};
Struct elev
{ char nume[21];
data dn;}
elev E1, E2;
```

3. Ce se va afișa în urma executării secvenței de program următoare? **(6 puncte)**

```
char a[3][12]={"bacalaureat", "la", "informatica"};
cout<<a[0]<<' '<<a[1][1]<<' '<<a[2][0];
```

SUBIECTUL III (30 de puncte)

1. Scrieți un program C/C++ care citește de la tastatură un număr natural impar N (N∈[2,50]) și elementele unui tablou bidimensional cu N linii și N coloane, numere întregi, apoi transformă tabloul în memorie, ștergând o linie și o coloană, la mijlocul său ca în exemplu. Tabloul obținut se afișează pe ecran, linie cu linie, elementele fiecărei linii fiind separate prin câte un spațiu. **(10 puncte)**

Exemplu: Dacă N=5 și tabloul este:

1	2	**3**	4	5
6	7	**8**	9	4
3	**4**	**5**	**6**	**7**
3	2	**3**	4	5
6	5	**7**	8	9

atunci se obține tabloul următor:

1	2	4	5
6	7	9	4
3	2	4	5
6	5	8	9

2. Subprogramul **numar** are doi parametri:
   - **n**, prin care primește un număr natural ($2 \le n \le 10^9$);
   - **m**, prin care furnizează un număr natural ($2 \le m \le 10^9$) cel mai mare număr natural care se poate obține folosind toate cifrele impare care apar în scrierea lui n, dacă în scrierea lui n nu apar cifre impare atunci m va avea valoarea **–1**. Scrieți în limbajul C/C++ definiția completă a subprogramului **numar**.

   Exemplu: Dacă **n=74317** atunci **m=7731**, dacă **n=246** atunci **m=-1**. **(10 puncte)**

3. Scrieți un program C/C++ care citește de la tastatură trei numere naturale n, m și k. Programul afișează în fișierul **bac.txt** numerele naturale cu exact k cifre care sunt divizibile atât cu n cât și cu m ($1 \le n$, $m \le 10^9$, ($1 \le k \le 9$).

   Exemplu: Dacă n = 15, m = 10, k = 2, atunci în fișier se vor afișa numerele 30 60 90.

   a) Se cere să se proiecteze un algoritm eficient din punct de vedere al timpului de executare și al spațiului de memorie utilizat și să se realizeze o descriere de 3-4 rânduri a algoritmului ales justificându-se eficiența acestuia. **(2 puncte)**

   b) Scrieți programul C/C++ corespunzător metodei descrise la **a)**. **(8 puncte)**

Teza 4

SUBIECTUL I (20 de puncte)

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

1. Valoarea variabilei **a** este un umăr natural. Care dintre expresiile C/C++ următoare are valoarea **1** dacă şi numai dacă valoarea variabilei **a are exact 3 cifre.**

a) a/1000==0
b) a/100!=0
c) a/1000==0&&a/100!=0
d) a/100==0&&a/10!=0

2. Un arbore cu rădăcină este reprezentat prin următorul vector de „taţi" **(6, 6, 5, 0, 6, 4, 4, 2),** determinaţi nodurile care pot fi alese ca rădăcină astfel încât nodul 6 să aibă un număr maxim de descendenţi.

a) 1, 2, 5 **b)** 3, 8 **c)** 4,7 **d)** 6

3. Se generează în ordine lexicografică toate şirurile de lungime 3 cu structura vocală – consoană – vocală formate din literele **a, b, c, d, e**. Ştiind că primele 3 şiruri generate sunt **aba, abe, aca,** care dintre următoarele secvenţe reprezintă o secvenţă de şiruri generate unul după altul?

a) eba, ebe, ece
b) eca, ebe, ece
c) eca, ece, ede
d) ebe, eca, ece

4. Care este numărul de grafuri neorientate cu 5 vârfuri cu proprietatea că vârfurile 1 şi 2, respectiv 1 şi 3 nu sunt adiacente?

a) 32 **b)** 4^3 **c)** 2^8 **d)** 2^6

5. Se consideră subprogramul **f** definit de mai jos.

```
int f (int n)
{
    if (n%10==5) return 2;
    else return 2*f(n+1);
}
```

Ce se va afişa în urma apelului **f(1)**?

a) 16 **b)** 32 **c)** 64 **d)** 8

SUBIECTUL II (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.

```
citește x
(x număr întreg)
y←0
pentru i←9,0,–1 execută
  cx←x
  cât timp cx>0 execută
    dacă cx%10=i atunci
      y← y*10+i
    ■
    cx←cx/10
  ■
■
scrie y
```

a) Scrieți valoarea care se va afișa pentru x= 7172. **(6 puncte)**

b) Scrieți două seturi distincte de valori pentru x astfel încât rezultatul afișat să fie 4332. **(6 puncte)**

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

d) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să se înlocuiască structura pentru cu alt tip de structură repetitivă. **(6 puncte)**

2. Se consideră declarările de mai jos, în care variabilele z memorează partea reală și partea imaginară a unui număr complex. Scrieți o secvență de instrucțiuni **C/C++** care citește de la tastatură informațiile despre numărul complex z și afișează pe ecran modulul numărului complex memorat în variabila z. **(6 puncte)**

```
struct complex
{ int pre;
int pim;}
complex z;
```

3. În secvența de instrucțiuni de mai jos variabilele i și j sunt de tip întreg, iar variabila A memorează un tablou bidimensional cu 5 linii și 5 coloane, numerotate de la 1 la 5. Elementele tabloului sunt numere întregi. Fără a utiliza alte variabile, scrieți una sau mai multe instrucțiuni care pot înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, tabloul memorat în variabila A să aibă elementele din figura de mai jos. **(6 puncte)**

```
for(i=1;i<=5;i++)
  for(j=1;j<=5;j++)
    ............
```

```
1 1 1 1 1
1 2 3 4 5
1 3 6 0 5
1 4 0 0 5
1 5 5 5 0
```

SUBIECTUL III (30 de puncte)

1. Scrieți un program C/C++ care citește de la tastatură un șir de caractere format din cel mult 100 de caractere (litere mici și mari ale alfabetului englez și caracterul spațiu) și afișează pe ecran șirul obținut prin inserarea unui număr minim de caractere #, astfel încât să nu existe în șir două caractere alăturate identice.
Exemplu: Dacă se citește șirul `copiii au acces` se va afișa `copi#i#i au ac#ces`
(10 puncte)

2. Subprogramul **numărare** are doi parametri:
   - **n**, prin care primește un număr natural ($1 \le n \le 10^9$);
   - **m**, prin care primește un număr natural ($1 \le m \le 10^9$, $n \le m$).

   Subprogramul returnează numărul de cifre utilizate pentru construirea numerelor naturale din intervalul `[n, m]`. Scrieți în limbajul C/C++ definiția completă a subprogramului **numărare**.
   Exemplu: Dacă **n = 7** și **m = 56** atunci, după apel, subprogramul va returna valoarea **115**.
   (10 puncte)

3. Scrieți un program C/C++ care citește din fișierul text **BAC.TXT** un șir cu cel mult 100 000 de numere întregi formate din cel mult 2 cifre fiecare și afișează pe ecran separate printr-un spațiu, numărul sau numerele din fișier cu număr maxim de apariții.
   Exemplu: Dacă în fișier sunt numerele `27 -8 43 27 -8 9 10 43` atunci pe ecran se vor afișa nu neapărat în această ordine `-8 43 27`.
   a) Se cere să se proiecteze un algoritm eficient din punct de vedere al timpului de executare și al spațiului de memorie utilizat și să se realizeze o descriere de 3-4 rânduri a algoritmului ales justificându-se eficiența acestuia. **(2 puncte)**
   b) Scrieți programul C/C++ corespunzător metodei descrise la punctul **a)**. **(8 puncte)**

Teza 5

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 reale, iar **a≤b.** Care dintre expresiile **C/C++** următoare are valoarea **1** dacă și numai dacă valoarea variabilei **z aparține** intervalului închis determinat de valorile variabilelor **a** și **b**?
   a) `(z>a||(z>b)`
   b) `(z<a)||(z>b)`
   c) `!(z<a && z>b)`
   d) `!(z<a || z>b)`

2. Un arbore cu rădăcină are exact 2019 noduri. Șiind că vectorul de tați T al acestui arbore are proprietatea că T[i]=[i/2], pentru oricare i de la 1 la 2019, atunci numărul de noduri care au exact un descendent este:
   a) 2 b) 2018 c) 1 d) 0

3. Se consideră algoritmul care determină toate parfumurile formate din 3 esențe din mulțimea {mosc, santal, tuberoze, liliac, lavandă}. Două parfumuri sunt diferite dacă diferă prin cel puțin o esență. Care este numărul de parfumuri care se pot realiza?
   a) 8 b) 12 c) 10 d) 9

4. Se consideră un graf orientat cu 7 noduri și 2 componente tare conexe, o componentă cu 4 noduri și o componentă cu 3 noduri. Numărul maxim de arce care pot exista în graf este:
   a) 14 b) 25 c) 30 d) 20

5. Se consideră subprogramul **f** definit mai jos.

```
int f (int n)
    {
        if (n==0) return 0;
        else if(n%3==0) return 1+f(n/10);
             else return f(n/10);
    }
```

Ce se va afișa în urma apelului f(23160)?
a) 2 b) 1 c) 3 d) 4

SUBIECTUL II (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.

```
citește x
(x număr natural)
cât timp x>9 execută
    y←0
    cât timp x>0 execută
        y←y+x%10
        x←[x/10]
    x←y
scrie x
```

a) Scrieți valoarea care se va afișa pentru x = 2019. **(6 puncte)**

b) Scrieți cel mai mic număr de două cifre și cel mai mare număr de două cifre care pot fi citite pentru x astfel încât rezultatul afișat să fie 9. **(6 puncte)**

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

d) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să nu se utilizeze structuri repetitive. **(6 puncte)**

2. Se consideră declarările de mai jos, în care variabila p memorează simultan numărul de vârfuri ale unui poligon (număr natural din intervalul $[3,10^2)$) și coordonatele vârfurilor acestuia (abscisa și ordonata) în sistemul de coordonate xOy (numere reale). Fără a utiliza variabile suplimentare scrieți o secvență de instrucțiuni **C/C++** care afișează pe ecran perimetrul poligonului memorat în variabila p. **(6 puncte)**

```
struct punct
{float x,y;};
struct poligon
{ int nr;
punct v[101];}p;
int i;
float per;
```

3. În secvența de instrucțiuni de mai jos variabilele **i** și **j** sunt de tip întreg, iar variabila **a** memorează un tablou bidimensional cu 5 linii și 5 coloane, numerotate de la 1 la 5. Elementele tabloului sunt numere întregi. Fără a utiliza alte variabile, scrieți una sau mai multe instrucțiuni care pot înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, tabloul memorat în variabila **a** să aibă elementele din figura de mai jos. **(6 puncte)**

```
for(i=1;i<=5;i++)
  for(j=1;j<=5;j++)
    ..............
```

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

SUBIECTUL III (30 de puncte)

1. Scrieți un program C/C++ care citește de la tastatură un șir de caractere format din cel mult 100 de caractere (litere mici și mari ale alfabetului englez) și afișează pe ecran pe aceeași linie separate prin spații șirurile obținute prin concatenarea prefixelor și sufixelor de aceeași lungime.
   Exemplu: Dacă se citește șirul de caractere **info** atunci pe ecran se va afișa **io info infnfo infoinfo.** **(10 puncte)**

2. Subprogramul **divizori** are un parametru **n**, prin care primește un număr natural ($2 \le n \le 10^4$). Subprogramul returnează suma divizorilor primi ai lui **n**. Scrieți în limbajul **C/C++** definiția completă a subprogramului **divizori**.
   Exemplu: Dacă **n = 12** atunci, după apel, subprogramul va returna valoarea **5,** divizorii primi ai lui 12 sunt 2 și 3. **(10 puncte)**

3. Scrieți un program C/C++ care citește din fișierul text **BAC.TXT** un șir S cu cel mult 100 000 de numere naturale formate din cel mult trei cifre fiecare. Asupra acestui șir se aplică în mod repetat următoarea prelucrare: se elimină din șir valorile prime, iar valorile neprime se incrementează cu valoarea 1. Prelucrarea se repetă până când în șir nu mai există niciun număr. Să se afișeze pe ecran de câte ori a fost aplicată această prelucrare.
   Exemplu: Dacă fișierul conține numerele 12 **11** 16 45 34 atunci după prima prelucrare vom avea valorile **13 17** 46 35; după a doua **47** 36; după a treia **37;** pe ecran va fi afișată valoarea **4.**
   a) Se cere să se proiecteze un algoritm eficient din punct de vedere al timpului de executare și al spațiului de memorie utilizat și să se realizeze o descriere de 3-4 rânduri a algoritmului ales justificându-se eficiența acestuia. **(2 puncte)**
   b) Scrieți programul C/C++ corespunzător metodei descrise la punctul **a).** **(8 puncte)**

Teza 6

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** și **b** memorează numere naturale. Care dintre expresiile C/C++ următoare are valoarea **1** dacă și numai dacă valorile variabilelor **a** și **b** au aceeași paritate?

```
1. (a+b)%2==0
2. (a-b)%2==0
3. a%2==b%2
4. a%2+b%2==0
```

a) 1, 2 **b)** 1, 2, 3, **c)** 2, 3, 4 **d)** 3, 4

2. Un graf neorientat cu 2019 vârfuri G = (X, U), unde X = {1, 2, ..., 2019}, are proprietatea că două vârfuri i, j din graf sunt adiacente dacă și numai dacă `i<j` și `j=2*i` sau `j=2*i+1`. Câte muchii are graful?

a) 1009 **b)** 2018 **c)** 2019 **d)** 2020

3. Într-o urnă sunt 5 bile albe și 3 bile negre. Care este numărul configurațiilor distincte care se pot obține prin extragerea bilelor din urnă și așezarea lor în linie în ordinea extragerii.

a) 54 **b)** 28 **c)** 120 **d)** 56

4. Se consideră un graf neorientat cu 12 noduri și 3 componenete conexe. Numărul maxim de muchii care pot exista în graf este:

a) 15 **b)** 20 **c)** 55 **d)** 45

5. Se consideră subprogramul f definit mai jos.

```
void f (int n)
{
    if(n)
    {
        for(int i=1; i<=n/2; i++)
            if(n%i==0)cout<<i;
        f(n-2);
    }
}
```

Ce se va afișa în urma apelului f(8)?

a) 12412312 **b)** 124123121 **c)** 1234121 **d)** 1241212

SUBIECTUL II (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.

```
citește x
(x număr întreg)
y←0
pentru i←2, x/2 execută
    dacă x% i=0 atunci
        y←y+1
dacă y=0 atunci
    scrie 1
altfel
    scrie y
```

a) Scrieți valoarea care se va afișa pentru x = 45 **(6 puncte)**

b) Scrieți toate numerele din intervalul [1, 9] care pot fi citite pentru x astfel încât să se afișeze valoarea 1. **(6 puncte)**

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

d) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să se înlocuiască structura pentru cu alt tip de structură repetitivă. **(6 puncte)**

2. Se consideră declarările de mai jos, în care variabilele A și B sunt coordonatele a două puncte (abscisa și ordonata) în sistemul de coordonate carteziene xOy (numere reale). Fără a utiliza variabile suplimentare scrieți o expresie C/C++ care are valoarea 1 dacă și numai dacă segmentul descris de punctele A și B se află pe prima bisectoare a sistemului de coordonate carteziene xOy. **(6 puncte)**

```
struct punct
{float x,y;}A, B;
```

3. În secvența de instrucțiuni de mai jos variabilele i și j sunt de tip întreg, iar variabila A memorează un tablou bidimensional cu 5 linii și 5 coloane, numerotate de la 1 la 5. Elementele tabloului sunt numere întregi. Fără a utiliza alte variabile, scrieți una sau mai multe instrucțiuni care pot înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, tabloul memorat în variabila A să aibă elementele din figura de mai jos. **(6 puncte)**

```
for(i=1;i<=5;i++)
  for(j=1;j<=5;j++)
    ..............
```

1	6	11	16	21
2	7	12	17	22
3	8	13	18	23
4	9	14	19	24
5	10	15	20	25

SUBIECTUL III (30 de puncte)

1. Scrieți un program C/C++ care citește de la tastatură un șir de caractere format din cel mult 100 de caractere (litere mici și spații) și afișează pe ecran lungimea maximă a unui cuvânt și numărul cuvintelor de lungime maximă din șir.

Exemplu: eu dau bacalaureat la informatica atunci pe ecran se va afișa **11 2.** **(10 puncte)**

2. Subprogramul **divizor** are doi parametri un parametru **n**, prin care primește un număr natural (**2≤n≤100**) și un parametru **a**, prin care primește un tablou unidimensional care memorează un șir de **n** numere naturale, fiecare cu cel mult **4** cifre. Subprogramul returnează cel mai mare divizor comun al celor n numere naturale din tablou. Scrieți în limbajul **C/C++** definiția completă a subprogramului **divizor**.
Exemplu: Dacă **n = 4** și tabloul memorează valorile **60 45 30 10** atunci, după apel, subprogramul va returna valoarea **5**. **(10 puncte)**

3. Un număr natural **N** este **p-compus**, dacă se poate scrie ca sumă de p numere naturale **consecutive**. Scrieți un program C/C++ care citește din fișierul text **BAC.TXT** de pe prima linie un număr natural p și de pe următoarele linii un șir S cu cel mult 100 000 de numere naturale cu cel mult 9 cifre fiecare, să se afișeze pe ecran pe aceeași linie separate printr-un spațiu, primul număr din sumă, dacă numărul din șir este p-compus sau mesajul NU în caz contrar.
Exemplu: Dacă fișierul conține numerele:
3
21 19 16 12 atunci pe ecran se va afișa **6 NU NU 3.**
a) Se cere să se proiecteze un algoritm eficient din punct de vedere al timpului de executare și al spațiului de memorie utilizat și să se realizeze o descriere de 3-4 rânduri a algoritmului ales justificându-se eficiența acestuia. **(2 puncte)**
b) Scrieți programul C/C++ corespunzător metodei descrise la **a)**. **(8 puncte)**

Teza 7

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, n memorează numere naturale. Care dintre expresiile C/C++ următoare are valoarea **1** dacă și numai dacă rădăcina pătrată a lui n nu depășește valoarea variabilei a și nici valoarea variabilei b?
   a) `sqrt(n)<=a&&sqrt(n)>=b`
   b) `sqrt(n)<=a&&sqrt(n)<=b`
   c) `sqrt(n)<=a||sqrt(n)<=b`
   d) `sqr(n)<=a&&sqr(n)<=b`

2. Un arbore cu n vârfuri este reprezentat prin matricea de adiacență. Numărul de elemente egale cu 1 din matricea de adiacență asociată grafului este:
   a) n **b)** $n^2 - n$ **c)** $n^2 - 2*n + 2$ **d)** $2*n - 2$

3. Un copil are în biblioteca personală 10 cărți în limba română, 4 cărți în limba germană și 6 cărți în limba engleză. Acesta trebuie să ducă la școală două cărți scrise în limbi diferite. Câte posibilități de alegere a cărților are?
   a) 120 **b)** 124 **c)** 240 **d)** 128

4. Fiind dat un graf orientat cu 10 noduri și fără circuite atunci numărul maxim de arce pe care le poate avea graful este:
   a) 110 **b)** 55 **c)** 45 **d)** 90

5. Se consideră subprogramul f definit mai jos.

```
void f (int n)
  {
      if(n)
       {
           f(n/3);
           cout<<n%3;
       }
  }
```

Ce instrucțiune de apel trebuie executată astfel încât în urma apelului să se afișeze 1012?
a) f(29) **b)** f(34) **c)** f(30) **d)** f(32)

SUBIECTUL II (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.

```
citește x, y
(x, y numere naturale)
r←x
cât timp y ≤ r execută
  r←r-y
■
k←(x-r)/y
scrie k, r
```

a) Scrieți valoarea care se va afișa pentru x = 140 și y = 15. **(6 puncte)**

b) Scrieți două valori care pot fi citite pentru x și y astfel încât valoarea variabilei k să fie 9. **(6 puncte)**

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

d) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să nu se utilizeze nicio structură repetitivă. **(6 puncte)**

2. Se consideră declarările de mai jos, în care variabilele A și B sunt coordonatele a două puncte (abscisa și ordonata) în sistemul de coordonate carteziene xOy (numere reale). Fără a utiliza variabile suplimentare scrieți o expresie C/C++ care are valoarea 1 dacă și numai dacă segmentul descris de punctele A și B se află pe una dintre axele sistemului de coordonate carteziene xOy. **(6 puncte)**

```
struct punct
{float x,y;}A, B;
```

3. În secvența de instrucțiuni de mai jos variabilele i și j sunt de tip întreg, iar variabila A memorează un tablou bidimensional cu 5 linii și 5 coloane, numerotate de la 1 la 5. Elementele tabloului sunt numere întregi. Fără a utiliza alte variabile, scrieți una sau mai multe instrucțiuni care pot înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, tabloul memorat în variabila A să aibă elementele din figura de mai jos. **(6 puncte)**

```
for(i=1;i<=5;i++)
  for(j=1;j<=5;j++)
    ..............
```

1	2	3	4	5
10	9	8	7	6
11	12	13	14	15
16	17	18	19	20
25	24	23	22	21

SUBIECTUL III (30 de puncte)

1. Scrieți un program C/C++ care citește de la tastatură două șiruri de caractere S și C de aceeași lungime, formate din cel mult 100 de caractere, șirul S este format din litere mici ale alfabetului englez, iar șirul C este format din cifre, programul codifică șirul S, înlocuind

fiecare literă S[i] din șir cu litera din alfabet aflată la distanța C[i], dacă șirul literelor se termină, se reia de la litera a. Șirul astfel obținut se afișează pe ecran.
Exemplu: Dacă se citesc șirurile **info 3212** atunci se va afișa șirul **lpgr.** **(10 puncte)**

2. Subprogramul **permuta** are trei parametri: un parametru n, prin care primește un număr natural ($2 \le n \le 100$), un parametru a, prin care primește un tablou unidimensional care memorează un șir de n numere naturale, fiecare cu cel mult 4 cifre și un parametru k prin care primește un număr natural ($1 \le k < n$). Subprogramul **permută circular la stânga** cu **k** poziții cele **n** numere naturale din tablou. Scrieți în limbajul C/C++ definiția completă a subprogramului **permuta**.
Exemplu: Dacă n = 4, k = 2 și tabloul memorează valorile 60 45 30 10 atunci, după apel tabloul va memora **30 10 60 45**. **(10 puncte)**

3. Scrieți un program C/C++ care citește din fișierul text **BAC.IN** un șir S cu cel mult 100 000 de numere naturale de forma $\mathbf{10^k}$, unde **k** este un număr natural din intervalul [0,9], programul afișează în fișierul **BAC.OUT** elementele șirului S în ordine crescătoare.
Exemplu: Dacă fișierul **BAC.IN** conține numerele 100 10 100 1 10 atunci în fișierul **BAC.OUT** vor fi afișate valorile **1 10 10 100 100.**
a) Se cere să se proiecteze un algoritm eficient din punct de vedere al timpului de executare și al spațiului de memorie utilizat și să se realizeze o descriere de 3-4 rânduri a algoritmului ales justificându-se eficiența acestuia. **(2 puncte)**
b) Scrieți programul C/C++ corespunzător metodei descrise la punctul **a)**. **(8 puncte)**

Teza 8

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 n, k, nr memorează numere naturale. Variabila `nr` trebuie să memoreze numărul de numere din intervalul `[0,n]` care sunt divizibile cu `k`. Cu ce expresie trebuie completată atribuirea `nr=...`?

a) `nr=n/k`
b) `nr=n/k+1`
c) `nr=n/k+n%k`
d) `nr=n%k+1`

2. Se dă graful neorientat complet cu n noduri, câte noduri va avea un arbore cu același număr de muchii ca graful dat?

a) $n*(n+1)/2$
b) $(n^2-n+2)/2$
c) $n*(n+1)/2+1$
d) $n*(n-1)/2-1$

3. La examenul de Bacalaureat la Subiectul I sunt cinci itemi de tip grilă, fiecare item având 4 variante de răspuns. În câte moduri pot fi completate răspunsurile la acest subiect, dacă există posibilitatea de a nu completa niciun răspuns?

a) 5^5 **b)** 4^5 **c)** 5^2 **d)** 4^2

4. Se consideră un graf neorientat cu 10 noduri și 2 componenete conexe, o componentă cu 7 noduri și o componentă cu 3 noduri. Numărul minim de muchii care pot exista în graf este:

a) 10 **b)** 8 **c)** 16 **d)** 14

5. Se consideră subprogramul f definit mai jos.

```
void f (int n)
{
        cout<<n;
        for(int i=1; i<=n/2; i++)
            if(n%i==0)f(i);
}
```

Ce se va afișa în urma apelului f(6)?

a) 6123121 **b)** 612312 **c)** 612131 **d)** 61213

SUBIECTUL II (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.

```
citește x
(x număr natural, x≠1)
y←0
pentru i←1, x/2 execută
    dacă x% i=0 atunci
        y←y+i
■
■
dacă y=x atunci
    scrie 1 ,' ', x
altfel
    scrie y
■
```

a) Scrieți valoarea care se va afișa pentru x = 12. **(6 puncte)**

b) Scrieți o valoare care poate fi citită pentru x astfel încât rezultatul afișat să fie 1,' ', x. **(6 puncte)**

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

d) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să se înlocuiască structura **pentru** cu alt tip de structură repetitivă cu test final. **(6 puncte)**

2. Se consideră declarările de mai jos, în care variabila t memorează simultan lungimile laturilor unui triunghi (numere reale pozitive). Fără a utiliza variabile suplimentare, scrieți o expresie **C/C++** care are valoarea 1 dacă și numai dacă trunghiul ale cărui laturi sunt memorate în variabila t este un triunghi isoscel, fără a fi triunghi echilateral. **(6 puncte)**

```
struct triunghi
{float a, b, c;} t;
```

3. În secvența următoare variabilele **n** și **i** sunt de tip întreg, iar variabila **s** permite memorarea unui cuvânt, șir de cel mult 10 de caractere. Cuvintele citite sunt formate din cifre zecimale și reprezintă numere de telefon, acestea sunt separate prin Enter.

```
n=.....;
for(i=1;i<=5;i++)
{
 cin>>s;
 ............
}
```

Fără a utiliza alte variabile, scrieți secvența de instrucțiuni care poate înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, variabila **n** să memoreze numărul de șiruri citite pentru care subșirul format din primele patru caractere ale lor coincide cu șirul 2019, iar acesta NU mai apare pe alte poziții în codul respectiv. **(6 puncte)**

Exemplu: Dacă se citesc șirurile, variabila **n** are valoarea 2.

2019745432
1378912217
2019345435
2019782019
1765412019

SUBIECTUL III (30 de puncte)

1. Scrieți un program C/C++ care citește de la tastatură un număr natural N (N∈[2,10]) și elementele unui tablou bidimensional cu N linii și N coloane, numere naturale din mulțimea{0, 1}, programul determină pentru fiecare linie din tablou numărul natural scris în baza 10 a cărui scriere în baza 2 pe N poziții binare este reprezentată de linia respectivă și afișează pe ecran cu spații între ele numerele determinate. **(10 puncte)**

   Exemplu: Dacă N=4 și tabloul este:
   ```
   0 1 0 1
   1 0 0 0
   1 0 0 1
   1 1 0 0
   ```
   atunci pe ecran se vor afișa valorile **5 8 9 12.**

2. Subprogramul **patrat** are un parametru **n**, prin care primește un număr natural ($2 \le n \le 10^9$). Subprogramul returnează cel mai mare pătrat perfect care este divizor al lui **n**. Scrieți în limbajul **C/C++** definiția completă a subprogramului **patrat**.
   Exemplu: Dacă **n = 72** atunci, după apel, subprogramul va returna valoarea **36**.
   (10 puncte)

3. Scrieți un program C/C++ care citește din fișierul text **BAC.TXT** un șir S cu cel mult 100 000 de numere naturale din intervalul [**2, 10^9**]. Pentru fiecare valoare x din șir se determină numărul de cifre egale cu zero de la sfârșitul lui x! (**unde x!=1*2*3...*x**, programul determină și afișează ecran cu spațiu între ele numărul de valori x din șir care au un număr maxim de valori egale cu zero la sfârșitul lui **x!** și care este această valoare maximă.
   Exemplu: Dacă fișierul conține numerele **102 12 50 100** atunci pe ecran vor fi afișate valorile **2 24.**
   a) Se cere să se proiecteze un algoritm eficient din punct de vedere al timpului de executare și al spațiului de memorie utilizat și să se realizeze o descriere de 3-4 rânduri a algoritmului ales justificându-se eficiența acestuia. **(2 puncte)**
   b) Scrieți programul C/C++ corespunzător metodei descrise la punctul **a)**. **(8 puncte)**

Teza 9

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 **x, y și z** au valori numere naturale de cel mult 4 cifre. Care dintre expresiile C/C++ următoare are valoarea **1** dacă și numai dacă exact una dintre valori este impară?
   a) `x%2 || y%2 || z%2`
   b) `(x+y)%2==0 || (x+z)%2==0 || (y+z)%2==0`
   c) `(x%10 + y%10 + z%10)%2>0`
   d) `x%10%2 + y%10%2 + z%10%2==1`

2. Numim înălțime a unui arbore cu rădăcină numărul de muchii ale celui mai lung lanț elementar care are una dintre extremități în rădăcina arborelui. Într-un arbore cu 9 noduri, toate frunzele sunt pe ultimul nivel și fiecare nod care nu este frunză, are cel mult trei fii. Care dintre următoarele variante poate fi vectorul de tați al unui astfel de arbore cu înălțimea minimă?
   T1 = (1, 2, 0, 3, 3, 2, 2, 3, 1),
   T2 = (0, 1, 2, 1, 2, 1, 4, 6, 4),
   T3 = (8, 8, 8, 1, 2, 2, 3, 9, 0),
   T4 = (6, 7, 6, 7, 0, 5, 5, 6, 7)

   a) T1, T2, T4 **b)** T2, T3 **c)** T2, T4 **d)** T2, T3, T4

3. Utilizând metoda backtracking, se generează în ordine lexicografică, toate modalitățile de a repartiza 5 studenți, în două amfiteatre A și B, astfel încât să fie cel mult 3 studenți într-un amfiteatru. Studenții sunt numerotați crescător de la 1 la 5, iar ordinea lor într-un amfiteatru nu contează. Primele 4 variante, în ordinea în care au fost generate, sunt: AAABB, AABAB, AABBA, AABBB. Care sunt cele două soluții ce vor fi generate imediat înainte și imediat după soluția BABAB?
   a) BABAA, BABBB **c)** BAABB, BABBA
   b) BABAA, BABBA **d)** BAABA, BBAAA

4. Matricea de adiacență a unui graf neorientat cu 20 de noduri, fără vârfuri izolate, conține 24 de valori egale cu 1. Care este numărul minim de componente conexe ale grafului?
   a) 8 **b)** 6 **c)** 10 **d)** 1

5. Se consideră subprogramul **f** definit mai jos.

```
int f(int x) {
    int n;
    if (x>0)
```

```
        { if(x%2==0) { cout<<x%10; n=1+f(x/10);}
          else { n=1+f(x/10); cout<<x%10;}
          return n;}
    else return 0;
}
```

Ce se va afișa în urma apelului `cout<<f(8172)`?

a) 2817 **b)** 27184 **c)** 28174 **d)** 8217

SUBIECTUL II (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.

citește n (n număr natural nenul)
a←1
b←[n / 2]
cât timp a ≠ 0 și b>0 execută
  c←n
  cât timp c>=b execută
    c←c – b
  ■
  a←c
  b←b – 1
■
b←b+1
scrie b

a) Scrieți valoarea care se va afișa pentru n = 39. **(6 puncte)**

b) Scrieți toate valorile de o cifră ce pot fi date pentru n astfel încât rezultatul afișat să fie 1. **(6 puncte)**

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

d) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să se utilizeze cel mult o structură repetitivă. **(6 puncte)**

2. Se consideră declarările C/C++ de mai jos.

```
struct produs  {
    int cod;
    float  cant,pret;
};
struct magazin {
    produs P[100];
    int nrProduse;
} M;
```

unde variabila **M** memorează produsele dintr-un magazin și nrProduse disponibile (maxim 100 produse). Scrieți o secvență de instrucțiuni C/C++ care afișează pentru magazinul M, pentru fiecare dintre cele nrProduse, pe câte o linie, valoarea totală a stocului (cantitate * preț), dacă acesta are cantitatea diferită de 0, respectiv codul produsului, în caz contrar. **(6 puncte)**

3. În secvența de instrucțiuni de mai jos variabilele **i** și **j** sunt de tip întreg, iar variabila **A** memorează un tablou bidimensional cu 6 linii și 6 coloane, numerotate de la 1 la 5. Elementele tabloului sunt numere întregi. Fără a utiliza alte variabile, scrieți una sau mai multe instrucțiuni care pot înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, tabloul memorat în variabila A să aibă elementele din figura de mai jos. **(6 puncte)**

```
for(i=0;i<=5;i++)
  for(j=5;j>=0;j--)
    ................
```

6	5	5	5	5	5
5	6	3	2	1	0
3	3	6	3	3	3
5	4	3	6	1	0
1	1	1	1	6	1
5	4	3	2	1	6

SUBIECTUL III **(30 de puncte)**

1. Scrieți un subprogram C/C++, denumit **circular**, care primește prin parametrii a și b două numere naturale, nenule, de cel mult 9 cifre. Subprogramul va returna cel mai mic număr de permutări circulare la dreapta, cu câte o poziție, ale cifrelor lui a, astfel încât să se obțină numărul b. Dacă nu e posibil, funcția va returna –1.
Exemplu: Pentru a = 120362 și b = 621203 subprogramul va returna 2, iar pentru a = 732 și b = 237 va returna –1. **(10 puncte)**

2. Spunem că două cuvinte A și B **rimează**, dacă sufixul lui A care începe cu ultima vocală coincide cu sufixul lui B care începe cu ultima vocală. De exemplu, cuvintele **savant** și **gigant** rimează, în timp ce sport și cert nu rimează. Scrieți un program C/C++ care citește de la tastatură, de pe prima linie S, un text de lungime maximă 255 caractere, litere mici și spații, iar de pe a doua linie, un cuvânt C, cu cel puțin 3 și cel mult 30 de litere, dintre care cel puțin una este vocală. Textul este format din cuvinte separate prin unul sau mai multe spații. Programul va afișa, pe linii diferite, cuvintele din text care rimează cu C sau mesajul NU EXISTA, dacă S nu conține astfel de cuvinte. **(10 puncte)**

De exemplu, dacă se citesc șirurile de mai jos
```
true is bool and school is cool
tool
```
se va afișa:
```
bool
school
cool
```

3. Un număr este **palindrom** dacă citind cifrele numărului de la dreapta la stânga și de la stânga la dreapta se obține același număr. De exemplu 12321, 2002 sunt palindroame, iar 12312 nu este palindrom.

Fișierul **text bac.txt** conține pe prima linie n, un număr natural nenul, mai mic decât 105, iar pe a doua linie un șir de n numere din intervalul [1, 9], separate prin câte un spațiu. Se cere afișarea pe ecran a celui mai mare palindrom par care se poate forma cu toate cifrele date. Dacă nu există, se va afișa –1. Proiectați un algoritm eficient din punct de vedere al memoriei utilizate și al timpului de executare.

Exemplu: Dacă fișierul **bac.txt** conține numerele

10

2 3 3 8 9 2 3 9 8 3 atunci pe ecran se afișează: 8933223398.

a) Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia. **(2 puncte)**

b) Scrieți programul C/C++ corespunzător algoritmului descris la punctul **a)**. **(8 puncte)**

Teza 10

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** au valori numere naturale nenule, **a ≤ b.** Care dintre următoarele instrucțiuni C/C++ atribuie variabilei întregi **c** numărul de valori pare din intervalul [a, b]?
   a) `c=(b-a+1)/2;`
   b) `c=a/2-b/2;`
   c) `c=b/2-(a-1)/2`
   d) `c=(a+b)/2;`

2. Într-un graf orientat cu 11 noduri, toate arcele sunt de forma (x,y), unde x și y au aceeași paritate și x<y. Care este numărul minim de arce de adăugat astfel încât graful să devină tare conex?
   a) 4 b) 2 c) 1 d) 0

3. Utilizând metoda backtracking, se generează toate cadourile formate din câte 3 obiecte distincte din mulțimea **{carte, tabletă, joc, stilou, ceas, patine}**, astfel încât să nu se găsească în același cadou **tabletă** și **joc** sau **carte** și **stilou.** Primele patru soluții obținute sunt, în această ordine: **(carte, tabletă, ceas) (carte, tabletă, patine), (carte, joc, stilou)** și **(carte, joc, patine).** Indicați a șasea soluție generată.
   a) **tabletă, joc, patine** b) **joc, stilou, ceas**
   c) **carte, stilou, ceas** d) **tabletă, stilou, ceas**

4. Se consideră următorul arbore dat prin vectorul de tați: T = (8, 9, 9, 9, 4, 4, 5, 5, 0, 5). Câte noduri pot fi alese ca rădăcină astfel încât înălțimea arborelui să fie maximă?
   a) 3 b) 5 c) 2 d) 6

5. Se consideră subprogramul f definită astfel:

```
void scrie(int a, int b){
  cout<<b;
  if(b){ scrie(b,a%b);
         cout<<a/b;}}
```

   Ce se va afișa la apelul `scrie(8,12)`?
   a) **12840210** b) **128421** c) **1248210** d) **1284210**

SUBIECTUL II (40 de puncte)

1. Se consideră algoritmul alăturat, descris în pseudocod.

S-a notat cu x % y restul împărțirii numerelor întregi x și y.

citește n (n număr natural)
d←0
pentru i←1, n execută
  citește x
  dacă d = 0 atunci
    d←x
  altfel
    repetă
      r←x%d
      x← d
      d←r
    până când r = 0
    d←x
scrie d

a) Scrieți valoarea care se va afișa dacă se citesc pe rând valorile 4, 8, 16, 10, 24. **(6 puncte)**

b) Pentru n = 3, dați exemplu de trei numere naturale, nenule și distincte ce pot fi citite în variabila x, pentru care algoritmul să scrie 1. **(6 puncte)**

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

d) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să înlocuiți structura „repetă până când" cu o structură repetitivă condiționată anterior. **(6 puncte)**

2. Se consideră declarările C/C++ de mai jos, în care variabila **E** memorează informațiile despre jucătorii unei echipe iar variabila **i** este de tip int. Scrieți o secvență de instrucțiuni C/C++ care, înlocuind punctele de suspensie, citește de la tastatură numărul de jucători ai echipei E și informațiile despre aceștia. **(6 puncte)**

```
struct Jucator {
   int nrTricou, marimeTricou;
};
struct Echipa
{ int nrJucatori ;
  Jucator juc[20];}E;

cin>>.....;
for(i=0;i<E.nrJucatori;i++)
      .....
```

3. În secvența de instrucțiuni de mai jos 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 5. Elementele tabloului sunt numere întregi. Fără a utiliza alte variabile, scrieți una sau mai multe instrucțiuni care pot înlocui punctele de suspensie astfel încât, în urma executării secvenței, tabloul memorat în variabila A să aibă elementele alăturat. **(6 puncte)**

```
for(i=0;i<6;i++)
  for(j=0;j<=2;j++)
    ............
```

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

SUBIECTUL III (30 de puncte)

1. Scrieți un subprogram C/C++, **divprim** cu trei parametri:
   – n, prin care primește un număr natural nenul de cel mult 5 cifre;
   – k, prin care primește numărul de elemente ale vectorului v;
   – v, vector de cel mult 100 de numere naturale nenule, ordonate crescător.
   Subprogramul va determina cel mai mic divizor prim al lui n și îl va căuta în vectorul v. Dacă divizorul prim nu va fi găsit, acesta va fi adăugat în v pe poziția corespunzătoare astfel încât ordinea crescătoare să se păstreze. Valorile modificate ale vectorului și dimensiunii acestuia vor fi furnizate tot prin parametrii k și v.
   Exemplu: Dacă la apel funcția va primi ca parametri n = 27, k = 4 și v = (2, 5, 11, 17), după execuția acesteia k = 5, iar v = (2, 3, 5, 11, 17). **(10 puncte)**

2. Numim **dublu-cuvânt** un cuvânt cu număr par de litere care este format prin dublarea unui cuvânt. De exemplu, `tictic` este un dublu-cuvânt, în timp ce `tictac` sau `abba` nu sunt. Scrieți un program C/C++ care citește de la tastură S, un text de lungime maximă 255 de caractere, format din cuvinte separate prin câte un spațiu și modifică în memorie șirul, înlocuind a doua jumătate a fiecărui dublu-cuvânt cu caracterul *. Dacă s-a făcut cel puțin o înlocuire, se va afișa pe ecran șirul rezultat, altfel se va afișa mesajul `NEMODIFICAT`.
   Exemplu: Dacă se citește: `se aude tictic tictac apoi dingding dingding`
   se va afișa: `se aude tic* tictac apoi ding* ding*` **(10 puncte)**

3. Fișierul text **bac.txt** conține cel mult 10^6 numere naturale de cel mult 9 cifre, separate prin câte un spațiu. Se cere afișarea pe ecran a valorii k pentru care intervalul de forma $[2^k, 2^{k+1})$ conține cele mai multe numere din șir. Dacă există mai multe astfel de intervale, se va afișa cel care are k maxim. Proiectați un algoritm eficient din punct de vedere al timpului de executare.
   Exemplu: Dacă fișierul bac.txt conține numerele
   `175 22 2018 512 1025 18` atunci pe ecran se afișează: `10`, deoarece sunt 2 intervale cu câte 2 numere: $22, 16 \in [2^4, 2^5)$, $2018, 1025 \in [2^{10}, 2^{11})$, dar al doilea interval are k maxim.
   a) Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia. **(2 puncte)**
   b) Scrieți programul C/C++ corespunzător algoritmului descris la punctul **a)**. **(8 puncte)**

Teza 11

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, c, d sunt întregi, a ≤ b, c ≤ d. Care dintre expresiile de mai jos are valoarea 1 dacă și numai dacă intersecția intervalelor (a, b) și [c, d] este vidă?

a) `c ≥ b && a < d`
b) `! (d > a || c ≤ b )`
c) `! (b > c && a < d)`
d) `a ≥ d || b < c`

2. Care este numărul de subgrafuri euleriene cu 3 vârfuri care se pot construi dintr-un graf neorientat cu 6 vârfuri și 6 valori egale cu 0 în matricea de adiacență corespunzătoare?

a) 30 **b)** 6 **c)** 15 **d)** 20

3. Se utilizează metoda backtracking pentru a genera în ordine lexicografică, toate anagramele distincte ale cuvântului **memorie** astfel încât să nu conțină litere identice pe poziții alăturate și litera o să apară după litera m, nu neapărat consecutiv. Primele soluții generate sunt **eiemrmo, eimemor, eimrmio, eimrmoi.** Care sunt penultima și ultima soluție generată?

a) `rmiemoe, rmimeoe`
b) `rimemeo, rimemoe`
c) `mrmieoe, mrmoeie`
d) `mirmeoe, rmiemoe`

4. Într-un arbore cu 100 de noduri, fiecare nod care nu este frunză are cel mult 2 fii. Care este numărul maxim de frunze ale acestuia?

a) 100 **b)** 51 **c)** 50 **d)** 99

5. Fie subprogramul F de mai jos.

```
int F (int n, int d)
{ if(d*d<n)
      return (n%d==0 ? d+n/d+F(n,d+1): F(n,d+1));
  else return d*d==n ? d: 0;}
```

Indicați apelul care determină calcularea sumei tuturor divizorilor proprii pozitivi ai numărului 100 (divizori diferiți de 1 și de 1 000).

a) F(100,1) **b)** F(100,2) **c)** F(2,100) **d)** F(1,100)

SUBIECTUL II (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.

```
citește n (n număr natural nenul)
k←0
a←1
cât timp a * a ≤ n execută
    dacă n % a=0 atunci
        b←[n /a]
        dacă a % 2=b % 2 atunci
            k←k + 1
    a←a+1
scrie k
```

a) Scrieți valoarea care se va afișa pentru n = 36. **(6 puncte)**

b) Scrieți cel mai mic număr de trei cifre ce poate fi citit pentru n astfel încât rezultatul afișat să fie 0. **(6 puncte)**

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

d) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să se utilizeze alt tip de structură repetitivă. **(6 puncte)**

2. Se consideră declarările C/C++ de mai jos: **Fisier**, care memorează numele, extensia și dimensiunea în octeți ale unui fișier de pe hard-disc și **Dosar**, care memorează numărul de fișiere dintr-un dosar (cel mult 100) și descrierea lor conform structurii Fisier. Scrieți o secvență de instrucțiuni C/C++ care atribuie variabilei sum dimensiunea totală a fișierelor din dosarul memorat în D, al căror nume începe cu litera A. **(6 puncte)**

```
struct Fisier {
 char nume[30],extensie[5];
 int dimensiune;};
struct Dosar {
  Fisier F[100];
  int nrFisiere;
} D;
int sum, i;
```

3. Ce se va afișa în urma executării secvenței de program de mai jos? **(6 puncte)**

```
char s[20]="bacalaureat", t[20]="mate-informatica";
s[3]=s[6]=' ';
strcpy(s+7,strchr(t,'i'));
cout<<s;
```

SUBIECTUL III (30 de puncte)

1. Scrieţi un program C/C++ care citeşte de la tastatură un număr natural N (N∈[2,50]), multiplu de 3 şi elementele unui tablou bidimensional cu cifre binare dispuse pe N linii şi N coloane, numerotate de la 0 la N – 1. Programul modifică tabloul în memorie înlocuind fiecare linie formată doar din valori 0 cu valorile reprezentând numărul de 1 de pe coloana corespunzătoare din tabloul iniţial apoi afişează rezultatul pe ecran. **(10 puncte)**

Exemplu: Dacă N=5 şi tabloul este:

```
1 0 0 1 1
1 1 1 0 1
0 0 0 0 0
1 1 0 1 1
0 0 0 0 0
```

atunci se obţine tabloul următor:

```
1 0 0 1 1
1 1 1 0 1
3 2 1 2 3
1 1 0 1 1
3 2 1 2 3
```

2. Subprogramul **nPrime** are trei parametri:
   - **n**, prin care primeşte un număr natural ($2 \le n \le 10^3$);
   - **k**, prin care furnizează numărul de valori prime cu n, mai mici decât n;
   - **p**, prin care furnizează şirul celor k valori prime cu n, în ordine crescătoare. Spunem că *x este prim cu n* dacă cele două numere nu au divizori comuni mai mari decât 1.

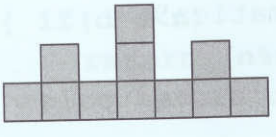
   Scrieţi în limbajul C/C++ definiţia completă a subprogramului **nPrime**.

   Exemplu: Dacă **n=20** atunci **k=8**, iar **p=(1,3,7,9,11,13,17,19).** **(10 puncte)**

3. Într-un depozit au fost aşezate cutii identice, una după alta, eventual suprapuse, astfel încât numărul maxim de cutii suprapuse într-o stivă este n ($0 < n \le 15$), iar între două stive cu acelaşi număr de cutii să existe cel puţin una cu mai multe cutii decât oricare dintre cele două. Considerăm că o stivă poate fi formată dintr-o singură cutie.

   Scrieţi un program care citeşte din fişierul text **bac.txt** o valoare n, reprezentând numărul maxim de cutii care pot fi stivuite una peste alta şi afişează pe ecran numărul maxim de stive care se pot construi respectând regulile date şi numărul de cutii ce ar putea fi aranjate astfel. Proiectaţi un algoritm eficient din punct de vedere al timpului de executare şi al spaţiului de memorie utilizat.

   Exemplu: Dacă n = 3, atunci numărul maxim de stive = 7 şi numărul total de cutii = 11, distribuite astfel:



   a) Realizaţi o descriere de 3-4 rânduri a algoritmului ales justificându-se eficienţa acestuia. **(2 puncte)**

   b) Scrieţi programul C/C++ corespunzător metodei descrise la punctul **a)**. **(8 puncte)**

Teza 12

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. Știind că x este o variabilă care memorează un număr întreg nenul, precizați care este cea mai mică valoare ce se poate obține la evaluarea următoarei expresii C/C++: `200%x - x%5`.

a) `-203` **b)** `0` **c)** `-195` **d)** `-4`

2. Care este numărul de valori egale cu 0 din matricea de adiacență a unui graf eulerian cu 10 vârfuri și număr maxim de muchii?

a) 20 **b)** 10 **c)** 15 **d)** 50

3. Se generează în ordine crescătoare toate numerele naturale de 4 cifre alese din mulțimea **3, 7, 2, 0, 5, 8** astfel încât cifrele de pe poziții alăturate să aibă parități diferite. Știind că primele 5 numere generate sunt **2305, 2307, 2385, 2387**, care dintre următoarele secvențe reprezintă o secvență de numere generate unul după altul:

a) 5087, 5237, 5238 **c)** 7850, 7852, 8305
b) 5087, 5237, 5238 **d)** 7830, 7832, 7835

4. Fie următoarea secvență de numere: 4, 3, 3, 1, 1, 1, 1, 1, 1. Care dintre următoarele șiruri poate fi vectorul de tați ai unui arbore cu secvența gradelor vârfurilor dată mai sus și înălțime maximă?

a) `(2,3,4,5,6,0,6,7,6)` **b)** `(0,1,1,1,2,2,9,9,1)`
c) `(9,1,1,1,2,2,3,3,0)` **d)** `(4,1,1,0,6,2,6,2,6)`

5. Se consideră subprogramul f definit mai jos.

```
void F(int a, int b)
{ if (b > a)
  {  cout << b;
     F(a, b/2);
  }
  else if (b < a)
       {  F(a/2, b);
          cout << a;
       }
       else cout << '#';
  }
```

Ce se va afișa în urma apelului F(7,8)?

a) `84237#` **b)** `842#37` **c)** `8#4237` **d)** `8#3742`

SUBIECTUL II (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.

```
citește a, b (a, b numere naturale, a≤b)
y←0
pentru x←b, a, –1 execută
    y←x
    k←0
    cât timp y>0 execută
        k←k+1–2*(y%2)
        y←[y/2]
    ■
    dacă k=0 atunci
        scrie x, ' '
    ■
■
scrie '*'
```

a) Ce se va afișa dacă a = 8 și b = 12? **(6 puncte)**

b) Pentru a = 25, care este cea mai mare valoare pe care o poate primi b astfel încât să se afișeze doar '*'? **(6 puncte)**

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

d) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să se înlocuiască „cât timp... execută" cu alt tip de structură repetitivă. **(6 puncte)**

2. Se consideră declarările C/C++ de mai jos, în care variabilele T memorează 100 de turnuri de forma paralelipiped, pentru care sunt cunoscute lungimea, lățimea, înălțimea și culoarea. Scrieți o secvență de instrucțiuni C/C++ care afișează culorile turnurilor care au formă de cub și sunt memorate în variabila T. **(6 puncte)**

```
struct Turn {
    int lungime, latime, inaltime;
    char culoare[20];};
Turn  T[100];
int i;
```

3. În secvența de instrucțiuni de mai jos variabilele **i** și **j** sunt de tip întreg, iar variabila **A** memorează un tablou bidimensional cu 6 linii și 6 coloane, numerotate de la 1 la 6. Elementele tabloului sunt numere întregi. Fără a utiliza alte variabile, scrieți una sau mai multe instrucțiuni care pot înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, tabloul memorat în variabila A să aibă elementele din figura de mai jos. **(6 puncte)**

```
for(i=1;i<=6;i++)
  for(j=1;j<=6;j++)
    ............
```

2	3	1	2	3	1
3	6	2	3	6	2
1	2	0	1	2	0
2	3	1	2	3	1
3	6	2	3	6	2
1	2	0	1	2	0

SUBIECTUL III (30 de puncte)

1. Scrieți un program C/C++ care citește de la tastatură un șir de caractere format din cel mult 250 de caractere, litere mici ale alfabetului englez și afișează pe ecran șirul rezultat și numărul minim de etape de eliminare astfel încât să nu mai existe în șir două caractere alăturate identice. O etapă de eliminare se aplică unei secvențe de caractere identice care începe la poziția i din șir, șterge secvența de la poziția i, apoi repetă ștergerea secvențelor de caractere identice ce includ poziția i, până cînd nu se mai obțin caractere identice alăturate.
Exemplu: Dacă se citește șirul `mtopppppootatnnnne` se va afișa `mate 2`, pentru că sunt 2 etape de eliminare: `toppppoot` și `nnnn`. **(10 puncte)**

2. Scrieți un subprogram C/C++, denumit **perechi**, care primește prin parametrii `n` și `m` două numere naturale, nenule, de cel mult două cifre. Subprogramul va furniza prin parametrul `k` numărul de perechi de numere `x, y (x<y<=n)`, care au cel mai mic multiplu comun `m`. Perechile determinate se vor afișa pe ecran, pe linii diferite, separate prin câte un spațiu.
Exemplu: Pentru n = 6 și m = 6 se afișează perechile:
```
1 6
2 3
2 6
3 6
```
, iar în `k` va fi valoarea `4`. **(10 puncte)**

3. Scrieți un program C/C++ care citește din fișierul text **BAC.TXT** un șir cu cel mult 1 000 000 de numere întregi formate dintr-o cifră fiecare și afișează pe ecran separate printr-un spațiu, numărul sau numerele din fișier cu număr maxim de apariții.
Exemplu: Dacă în fișier sunt numerele `1 -8 1 1 -8 -8 9 9` atunci pe ecran se vor afișa nu neapărat în această ordine `-8 1`.
   a) Se cere să se proiecteze un algoritm eficient din punct de vedere al timpului de executare și al spațiului de memorie utilizat și să se realizeze o descriere de 3-4 rânduri a algoritmului ales justificându-se eficiența acestuia. **(2 puncte)**
   b) Scrieți programul C/C++ corespunzător metodei descrise la punctul **a)**. **(8 puncte)**

Teza 13

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 **x, y** și **z** sunt de tip întreg. Care dintre următoarele expresii C/C++ are valoarea 1 dacă și numai dacă cel mult două dintre variabilele precizate au valori egale?
   a) `!(x==y && x==z)||y!=z` c) `(x-y)*(x-z)*(y-z)==0`
   b) `x==y || x==z || y==z` d) `x==y==z`

2. Care este înălțimea maximă a unui arbore cu 50 de noduri în care fiecare nod care nu este frunză are mai mulți fii decât părintele său?
   a) 25 b) 10 c) 9 d) 11

3. Se generează prin metoda backtracking mulțimi distincte cu elemente numere naturale nenule și cu proprietatea că suma elementelor fiecărei mulțimi este egală cu 7 astfel: `{1,2,4},{1,6},{2,5},{3,4},{7}`. Folosind aceeași metodă pentru a genera mulțimi distincte cu elemente numere naturale nenule și cu proprietatea că suma elementelor fiecărei mulțimi este egală cu 9, stabiliți în ce ordine sunt generate următoarele mulțimi:
   `M1= {2, 3, 4}; M2={3, 6}; M3={2, 7}; M4={1, 8}.`
   a) M4, M1, M3, M2 c) M1, M3, M2, M4
   b) M4, M1, M2, M3 d) M1, M2, M3, M4

4. **Distanta** dintre două noduri într-un graf neorientat este lungimea minimă a unui lanț cu extremitățile în cele două noduri. **Ponderea unui nod** este suma distanțelor de la acesta la toate celelalte noduri din graf. Care poate fi ponderea maximă a unui nod într-un graf neorientat cu 10 noduri?
   a) 20 b) 90 c) 10 d) 45

5. Se consideră subprogramul F definit de mai jos.

```
void F()
{  int x; cin>>x;
   if (x>0) { cout<<x<<' ';F();}
 else if (x<0) { F(); cout<<x<<' ';}
}
```

Ce se va afișa la citirea următorului șir de valori: `4 -5 -2 8 7 0`?
a) `7 8 4 -2 -5` c) `-5 -2 4 8 7`
b) `4 8 7 -5 -2` d) `4 8 7 -2 -5`

SUBIECTUL II (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.

citește n (n număr natural)
x←0; y←1;
m←[n/2];
cât timp m>0 execută
  x←x+y
  y←y+x
  m←m–1
■
z ← (1–n%2)* x+(n%2)*y
scrie z

a) Scrieți valoarea care se va afișa pentru n = 7. **(6 puncte)**

b) Scrieți cea mai mică valoare care se poate citi pentru n astfel încât rezultatul afișat să fie un număr de trei cifre. **(6 puncte)**

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

d) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să se înlocuiască structura repetitivă cât timp... execută cu o structură repetitivă cu condiție finală. **(6 puncte)**

2. Variabila C memorează un cerc descris prin raza R, un număr natural nenul și O, un punct de coordonate reale a, b reprezentând abscisa și ordonata centrului acestuia. Se consideră declarările C/C++ corespuzătoare descrierii de mai jos.

```
struct Punct {
  float x,y;};
struct Cerc {
  Punct centru;
  int raza;} C;
```

Scrieți o expresie C/C++ care să aibă valoarea 1 dacă și numai dacă cercul C are centrul pe axa Ox și intersectează axa Oy. **(6 puncte)**

3. În secvența de instrucțiuni de mai jos variabilele **i** și **j** sunt de tip întreg, iar variabila **A** memorează un tablou bidimensional cu 5 linii și 5 coloane, numerotate de la 1 la 5. Elementele tabloului sunt numere întregi. Fără a utiliza alte variabile, scrieți una sau mai multe instrucțiuni care pot înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, tabloul memorat în variabila **A** să aibă elementele din figura de mai jos. **(6 puncte)**

```
for(i=5;i>=1;i--)
  for(j=1;j<6;j++)
    ..............
```

5	1	1	1	2
5	2	0	0	1
5	1	2	0	1
5	2	1	2	1
5	5	5	5	5

SUBIECTUL III (30 de puncte)

1. Se consideră un text cu cel mult 200 de caractere, litere, cifre și spații. Cuvintele sunt formate din litere ale alfabetului englez și/sau cifre și sunt separate prin câte un spațiu. Scrieți un program C/C++ care citește de la tastatură un text de tipul menționat mai sus și construiește în memorie apoi afișează pe ecran șirul format din prima literă a fiecarui cuvânt care nu conține nicio cifră, transformată în majusculă. Dacă nu există astfel de cuvinte, se afișează pe ecran mesajul **nu există**.
   Exemplu: Pentru șirul `Varianta T100 data in 10nov la simulare 12C` se afișează `VDILS`. **(10 puncte)**

2. Un număr n se numește **rotund** dacă este par și în reprezentarea sa matematică în baza 2, nu sunt două cifre egale alăturate. De exemplu, numărul `10` este un număr rotund deoarece reprezentarea sa în baza 2 este `1010`. Numărul `18` nu este rotund deoarece reprezentat în baza 2 este `10010`.
   Scrieți definiția completă a unui subprogram **rotund**, care primește prin intermediul parametrului n un număr natural nenul de cel mult 9 cifre și returnează 1 dacă n este număr rotund și 0 altfel. **(10 puncte)**

3. Fiind date două numere a și b, îl numim pe **a sufix al lui b** dacă a este egal cu b sau dacă b se poate obține din a prin alipirea la stânga a unor noi cifre. Exemplu: 12 este sufix al lui 12, iar 15 este sufix al lui 31415. Fișierul **bac.txt** conține pe prima linie un număr natural x, cu cel mult 4 cifre, iar pe a doua linie un șir de cel puțin două și cel mult 1 000 000 de numere naturale cu cel mult nouă cifre. Numerele din șir sunt separate prin câte un spațiu.
   Se cere să se afișeze pe ecran ultimul termen al șirului care are ca sufix numărul x. Dacă în șir nu există o astfel de valoare, pe ecran se afișează mesajul **nu exista**. Pentru determinarea numărului cerut se utilizează un algoritm eficient din punctul de vedere al memoriei și al timpului de executare.
   Exemplu: Dacă fișierul **bac.txt** conține numerele
   `12`
   `3445 89312 1212 12 67120 312 1234578` atunci pe ecran se afișează `312`.

   a) Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia. **(2 puncte)**
   b) Scrieți programul C/C++ corespunzător algoritmului descris la punctul **a)**. **(8 puncte)**

Teza 14

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 **c** memorează numere naturale nenule. Care dintre expresiile C/C++ următoare este 1 dacă și numai dacă c este un multiplu comun al valorilor memorate în a și b?
   a) `a%c+b%c==0` **c)** `!(a%c==0 || c%b==0)`
   b) `(c%a)*(c%b)==0` **d)** `c%a+c%b==0`

2. Un graf orientat cu 5 vârfuri, numerotate de la 1 la 5, are arcele `(1,4)`, `(1,5)`, `(2,1)`, `(2,3)`, `(3,1)`, `(3,4)`, `(4,2)`, `(5,1)`, `(5,3)`. Numărul maxim de arce care se pot elimina, astfel încât graful parțial obținut să fie tare conex este:
   a) 2 **b)** 3 **c)** 4 **d)** 5

3. Cu ajutorul metodei backtracking se generează în ordine lexicografică, toate parolele formate din 5 caractere distincte din mulțimea `{'a','b','c','e','f','g','u'}` și care conțin cel puțin 2 vocale și cel puțin 2 consoane. Primele soluții generate sunt `abcef`, `abceg`, `abceu`, `abecf`. Care dintre șirurile următoare NU reprezintă secvență de șiruri generate consecutiv?
   a) `fegua, fegub, feguc` **c)** `bacue, bacuf, bacug`
   b) `gufab, gufeb, gufec` **d)** `efabu, efacb, efacg`

4. Se consideră un graf neorientat conex cu 100 de muchii. Numărul maxim de noduri din graf este:
   a) 100 **b)** 99 **c)** 101 **d)** 50

5. Se consideră subprogramul F definit mai jos.

```
long F(int x, int y)
{ if (x==y || x==0)
    return 1;
  else{ y--;
    return F(x,y) + F(x-1,y);}
}
```

Ce se va afișa în urma apelului `F(7,10)`?
a) 120 **b)** 100 **c)** 75 **d)** 34

SUBIECTUL II (40 de puncte)

1. Se consideră algoritmul alăturat, descris în pseudocod.

```
citește n (n număr natural nenul)
k←1; p←0
cât timp n > 0 execută
    citește x (x, număr întreg)
    dacă x < 0 și p > x atunci
        p←x
    altfel
        dacă x<0 && p<=x atunci
            k←0
    n←n – 1
scrie k
```

a) Scrieți valoarea care se va afișa dacă se citesc în ordine numerele 6 3 –2 5 –4 –8 1. **(6 puncte)**

b) Dacă pentru n se citește valoarea 4, scrieți un șir de 4 valori distincte astfel încât rezultatul afișat să fie 0. **(6 puncte)**

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

d) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să se înlocuiască structura repetă „cât timp... execută" cu o structură repetitivă „pentru". **(6 puncte)**

2. Considerăm că un produs este descris prin cod (număr natural), luna și anul expirării (numere naturale) și prețurile a 5 posibili furnizori (numere reale). Scrieți declararea C/C++ a unei structuri cu eticheta **Produs** și a unei variabile **P** capabilă să memoreze 100 de produse, astfel încât construcțiile de mai jos să fie corecte sintactic. **(6 puncte)**

```
P[0].cod
P[0].data_expirarii.luna,  P[0].data_expirarii.an
P[0].pret_furnizor[1]
```

3. În secvența de instrucțiuni de mai jos variabilele **i** și **j** sunt de tip întreg, iar variabila **A** memorează un tablou bidimensional cu 7 linii și 7 coloane, numerotate de la 1 la 7. Elementele tabloului sunt caractere. Fără a utiliza alte variabile, scrieți una sau mai multe instrucțiuni care pot înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, tabloul memorat în variabila **A** să aibă elementele din figura de mai jos. **(6 puncte)**

```
for(i=0;i<=6;i++)
  for(j=0;j<=6;j++)
    ..............
```

```
* - - - - - *
* * - - - * *
* - * - * - *
* - - * - - *
* - - - - - *
* - - - - - *
* - - - - - *
```

SUBIECTUL III (30 de puncte)

1. Operația **WRAP de dimensiune** n, aplicată asupra unui text format din cuvinte separate prin spații, va încadra textul pe un număr minim de linii de n caractere, astfel încât orice cuvânt va fi afișat complet pe linia sa. Dacă la sfârșitul liniei rămân caractere ce nu pot fi ocupate de un cuvânt ele se vor înlocui cu *. Dacă un spațiu din text ajunge pe prima poziție din linie, el se va ignora, deoarece nicio linie nu poate începe cu spațiu.
Scrieți un program C/C++ care citește de la tastatură, de pe prima linie, un număr natural n ($10 < n < 50$), iar de pe a doua linie, un text S de cel mult 200 de caractere (litere mici ale alfabetului englez și spații), format din cuvinte separate prin câte un spațiu. Textul nu începe și nu se termină cu spații. Programul afișează pe ecran liniile paragrafului în care se încadrează textul după operația WRAP de dimensiune n. Dacă afișarea nu e posibilă se va scrie pe ecran mesajul IMPOSIBIL. **(10 puncte)**
De exemplu, dacă se citește **n = 20** și textul:

```
bacul la info este usor daca esti foarte bine pregatit
se va afișa
bacul la info este *
usor daca esti *****
foarte bine pregatit
```

2. Scrieți în limbajul **C/C++** definiția completă a subprogramului **regulat** care primește ca parametru un număr natural nenul n, de cel mult 5 cifre și returnează cel mai mare număr regulat, mai mic sau egal cu n. Un număr natural nenul se numește **regulat** dacă poate fi scris sub forma $2^A * 3^B * 5^C$, unde A, B, C sunt numere naturale.
Exemplu: Pentru **n = 44**, subprogramul va returna după apel valoarea **40**, $40 = 2^3 * 3^0 * 5^1$. **(10 puncte)**

3. Din fișierul text **BAC.TXT** se citesc, de pe prima linie un număr k ($k < 10^6$), iar de pe linia următoare cel mult 1 000 000 de numere naturale cu cel mult 2 cifre fiecare, separate prin câte un spațiu. Să se afișeze pe ecran numărul de pe a doua linie din fișier care va fi pe poziția k în șirul ordonat descrescător, dacă există sau –1 în caz contrar. Să se proiecteze un algoritm eficient din punct de vedere al timpului de executare.
Exemplu: Dacă fișierul conține numerele:
6
21 9 16 2 16 2 9 4 9 21 9 2 8 atunci pe ecran se va afișa **9**.
a) Realizați o descriere de 3-4 rânduri a algoritmului ales justificând eficiența acestuia. **(2 puncte)**
b) Scrieți programul C/C++ corespunzător metodei descrise la punctul **a)**. **(8 puncte)**

Teza 15

SUBIECTUL I (20 de puncte)

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

1. Variabila a memorează un număr natural cu 4 cifre. Care dintre expresiile C/C++ următoare inserează cifra 0 în numărul n, pe poziţia cifrei sutelor?

a) `n=(n/100)*10000 + n%100`
b) `n=n%10+n/10*10+n/100*1000`
c) `n=n/10%100*1000+n/100`
d) `n=n%100+1000*n/100`

2. Un graf orientat cu 5 noduri numerotate de la 1 la 5, este dat prin listele de adiacenţă de mai jos.

L1: 2,5 L2: 5 L3: 2 L4: 1 L5: 1,2,4

Care este numărul minim de arce ce trebuie adăugate astfel încât graful orientat să fie complet?

a) 2 **b)** 12 **c)** 5 **d)** 10

3. Se generează cu ajutorul metodei backtracking, în ordine crescătoare, numere formate doar din cifrele 0, 1 şi 2, care au suma cifrelor egală cu 2. Primele 5 numere, în ordinea în care au fost generate, sunt: 2, 11, 20, 101, 110. Dacă se generează după aceeaşi regulă numere formate doar din cifrele 0, 1, 2 şi 3, care au suma cifrelor egală cu 4, care vor fi primele 3 astfel de numere scrise?

a) 13, 31, 103, 130, 301
b) 13, 22, 31, 103, 112
c) 4, 13, 22, 31, 40
d) 13, 22, 31, 40, 103

4. Într-un graf neorientat cu 10 de vârfuri, suma gradelor vârfurilor este 40. Care este numărul maxim de componente conexe dintr-un astfel de graf?

a) 5 **b)** 4 **c)** 6 **d)** 1

5. Se consideră subprogramul f definit alăturat:

```
void F(int a, int b){
        for(int i=a; i<=b; i++){
                F(i+1, b);
                cout<<i;
        }
}
```

Care instrucţiune de apel trebuie executată astfel încât în urma apelului să se afişeze F(1,3)?

a) 1323323 **b)** 32313 **c)** 32123 **d)** 3231323

SUBIECTUL II (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.

citește n (n număr natural, n>1)
p←1; m←0; x←2
cât timp n >= x execută
  dacă n%x=0 atunci
    m←m+1; n←[n/x]
  altfel
    dacă m%2 ≠ 0 atunci
      p←p*x
    m←0; x←x+1
dacă m%2 ≠ 0 atunci
  p←p*x
scrie p

a) Scrieți valoarea care se va afișa pentru n = 23100. **(6 puncte)**

b) Scrieți cel mai mic și cel mai mare număr de două cifre care poate fi citit pentru n astfel încât algoritmul să afișeze valoarea 1. **(6 puncte)**

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

d) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să se înlocuiască structura repetitivă „cât timp... execută" cu o structură repetitivă cu condiție finală. **(6 puncte)**

2. Pentru memorarea rezultatelor la examenul de BAC obținute de un candidat se consideră declarările C/C++ de mai jos.

```
struct rezBac {
  int mat,rom,inf;
  char rez;
} candidat;
```

Scrieți o instrucțiune C/C++ care să atribuie litera **'A'** rezultatului elevului **candidat**, dacă fiecare dintre notele la probele de română, mate și info sunt cel puțin 5, iar media lor este cel puțin 6, iar în caz contrar va atribui litera **'R'**. **(6 puncte)**

3. În secvența de instrucțiuni de mai jos variabilele **i** și **j** sunt de tip întreg, iar variabila **A** memorează un tablou bidimensional cu 6 linii și 6 coloane, numerotate de la 1 la 6. Elementele tabloului sunt numere întregi. Fără a utiliza alte variabile, scrieți una sau mai multe instrucțiuni care pot înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, tabloul memorat în variabila **A** să aibă elementele din figura de mai jos. **(6 puncte)**

```
for(i=1;i<=3;i++)
  for(j=1;j<=3;j++)
    ..............
```

```
1 1 1 1 1 1
1 2 2 2 2 1
1 2 3 3 2 1
1 2 3 3 2 1
1 2 2 2 2 1
1 1 1 1 1 1
```

SUBIECTUL III (30 de puncte)

1. Scrieți un program C/C++ care citește de la tastatură un șir S, format din cel mult 100 de caractere, litere mici ale alfabetului englez. Programul modifică în memorie șirul S, adăugând la sfârșit un număr minim de caractere astfel încât S să fie palindrom (parcurs de la stânga la dreapta și de la dreapta la stânga, să se obțină același șir), apoi afișează pe ecran șirul obținut după transformare.
Exemplu: Dacă se citește șirul **repetate** atunci se va afișa șirul **repetateper,** iar dacă se citește șirul **info** atunci se va afișa șirul **infoofni.** **(10 puncte)**

2. Subprogramul **compact** primește prin parametrul n, un număr natural ($2 \le n \le 1000$) și prin parametrul a, un tablou unidimensional care memorează un șir de n numere naturale, fiecare cu cel mult 2 cifre. Subprogramul elimină din tablou, pentru fiecare a_i, $1 \le i \le n$, elementele egale cu acesta, situate pe poziții consecutive, astfel încât, tabloul să conțină la final, un număr maxim de elemente distincte, în ordinea în care au apărut în șirul primit ca parametru. Șirul modificat și numărul de elemente rămase după eliminare vor fi furnizate tot prin intermediul parametrilor n și a. Scrieți în limbajul C/C++ definiția completă a subprogramului **compact**.
Exemplu: Dacă n=9 și tabloul memorează valorile 1 2 2 2 3 3 5 8 9 atunci, după apel n=6, iar tabloul va memora 1 2 3 5 8 9. **(10 puncte)**

3. Un șir de numere $x_1, x_2, \ldots x_k, \ldots x_m$ conține o **secvență – V** dacă are un element x_k ($1<k<m$) și există **i** și **p** ($1 \le i<k<p \le m$) cu proprietatea: $x_i > x_{i+1} > \ldots > x_k$ și $x_k < x_{k+1} < \ldots < x_p$. Scrieți un program care citește din fișierul **bac.txt** cel mult 1 000 000 numere întregi, având cel mult nouă cifre și afișează pe ecran lungimea maximă a unei secvențe – V și numărul de secvențe – V de lungime maximă pentru șirul de numere din fișier sau afișează mesajul **Nu există**, dacă nu există nicio secvență – V în șir. Elaborați un algoritm eficient ca timp de execuție și spațiu de memorie utilizat.
Exemplu: Pentru n = 14, șirul de numere 9,8,**2**,11,14,10,**9**,20,19,14,**9**,20,56,17 conține două secvențe – V cu lungimea maximă egală cu 5 și se va afișa pe ecran 5 2.
a) Descrieți în limbaj natural algoritmul proiectat și justificați eficiența acestuia. **(2 puncte)**
b) Scrieți programul C/C++ corespunzător algoritmului descris. **(8 puncte)**

Teza 16

SUBIECTUL I **(20 de puncte)**

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

1. Variabilele a şi b memorează numere întregi. Care dintre următoarele expresii C/C++ are valoarea 1 dacă şi numai dacă a şi b sunt numere impare, de semne contrare?

a) `a%2+b%2==2 && a<0 && b>0`
b) `a*b<0 && (a*b)%2!=0`
c) `(a<0 && b>0||a>0 && b<0) && a%2!=b%2`
d) `a%2 && b%2 && a+b<0`

2. Un graf neorientat cu 10 noduri este format din 3 componente conexe, fiecare subgraf complet. Care dintre afirmaţiile următoare sunt adevărate?

A1. gradul maxim al unui nod poate fi 7 **A2.** graful nu poate conţine vârfuri izolate
A3. suma gradelor vârfurilor e impară **A4.** graful are cel puţin 12 muchii

a) A1 şi A2 **b)** A2, A3 şi A4 **c)** A1 şi A3 **d)** A2 şi A4

3. Se consideră generate cu ajutorul metodei backtracking toate rezultatele posibile ale unei serii de 10 aruncări cu banul care produce secvenţa $R_1R_2 \ldots R_{10}$ unde $R_k \in \{C, P\}$ este rezultatul aruncării k (C – cap, P – pajură). Câte secvenţe vor avea acelaşi număr de capete şi de pajuri?

a) 1024 **b)** 512 **c)** 252 **d)** 32

4. Într-un graf orientat cu **7** vârfuri, numerotate de la **1** la **7**, pentru oricare două vârfuri ale sale **i** şi **j** (**i≠j**), există arcul **(i, j)** fie dacă **i** este multiplu al lui **j**, fie dacă **i** şi **j** au aceeaşi paritate, iar **i<j**. Care sunt vârfurile pentru care gradul interior este mai mare sau egal cu cel exterior?

a) 1, 5, 7 **b)** 2, 3, 6 **c)** 1, 7 **d)** 4, 5

5. Se consideră subprogramele F şi G definite mai jos.

```
int F(int a){
    if(a%2==0) return F(a/2);
    else return a;
}
void G(int n)
{
    if (n<1) return 1;
    else return F(n*G(n-1));
}
```

Ce se valoare se obţine prin apelul G(6)?

a) 120 **b)** 720 **c)** 24 **d)** 45

SUBIECTUL II (40 de puncte)

1. Se consideră algoritmul alăturat, descris în pseudocod.

citește n, x (n, x numere naturale)
pentru i←1, n – 1 execută
  citește y (x,y numere întregi, x ≤ y)
  dacă x ≤ 99 și y ≤ 10 atunci
    k←k+1
  ■
  x←y
■
scrie k

a) Scrieți valoarea care se va afișa dacă se citesc în ordine numerele 5 8 9 35 256 1000. **(6 puncte)**

b) Pentru n = 4, scrieți 4 valori distincte ce pot fi citite în x și y astfel încât rezultatul afișat să fie 0. **(6 puncte)**

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

d) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să se înlocuiască structura „pentru... execută" cu o structură repetitivă cu condiție inițială. **(6 puncte)**

2. Considerăm că o campanie de reduceri este descrisă prin titlu (maxim 30 de caractere), luna și ziua de început, durata exprimată în zile consecutive și procentul de discount (numere naturale nenule). Scrieți declararea C/C++ a unei structuri cu eticheta **Campanie** și a unei variabile **C**, capabilă să memoreze 50 de campanii, astfel încât construcțiile de mai jos să fie corecte sintactic. **(6 puncte)**

```
C[1].titlu
C[1].start.luna,  C[1].start.an
C[1].procent
```

3. În secvența următoare variabilele `n` și `i` sunt de tip întreg, iar variabila `s` permite memorarea unui cuvânt format din cel mult 200 de caractere. Cuvântul citit este format din litere mici. Fără a utiliza alte variabile, scrieți secvența de instrucțiuni care poate înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, afișează, în ordine crescătoare a lungimilor, sufixele lui `s` care încep cu cuvântul **`info`**. **(6 puncte)**

```
cin>>s;
n=.....;
for(i=0;i<n;i++)
{
 ............
}
```

SUBIECTUL III (30 de puncte)

1. Scrieți un program C/C++ care citește de la tastatură un număr natural N (N∈[2,10]) și elementele unui tablou bidimensional cu N linii și N coloane, caractere din mulțimea {'X', 'U'}. Programul afișează DA, dacă există cel puțin o linie, o coloană sau o diagonală completată doar cu caracterul 'X' și NU, altfel. **(10 puncte)**

   Exemplu: dacă N=4 și tabloul este:
   X U U X atunci pe ecran se va afișa DA
   U X X X
   X X U X
   X U X X

2. Subprogramul **nMax** primește prin parametrul n un număr natural de cel mult 6 cifre și prin parametrul c o cifră nenulă. Subprogramul furnizează tot prin parametrul n numărul maxim care se poate obține inserând în n cifra c. Scrieți în limbajul C/C++ definiția completă a subprogramului **nMax**.

   Exemplu: Dacă **n = 57332** și **c = 4** atunci, după apel, subprogramul va returna valoarea **574332**. **(10 puncte)**

3. Numim **interval asociat** unui șir de numere, perechea de numere naturale a,b (a≤b) cu proprietatea că șirul este format din toate valorile naturale distincte, cuprinse între a și b și scrise în ordine crescătoare. De exemplu, șirul 4 5 6 7 8 are intervalul asociat 4 8, iar șirul 1 2 3 5 7 8 9 este format din trei subșiruri ale căror intervale asociate sunt 1 3, 5 5 și 7 9.
   Scrieți un program C/C++ care citește din fișierul text **BAC.TXT** un șir S cu cel mult 1 000 000 de numere naturale din intervalul [0,10^3] și afișează pe ecran extremitățile intervalului care reprezintă intersecția tuturor intervalelor asociate, de lungime maximă, din care este format șirul dat sau mesajul mulțimea vidă, în cazul în care acestea nu conțin niciun element comun.

   Exemplu: Dacă fișierul conține numerele 2 3 4 1 2 3 4 5 6 3 4 5 0 1 2 3 atunci pe ecran vor fi afișate valorile 2 3 deoarece șirul este format din 4 intervale asociate 2 4, 1 6, 3 5 și 0 3 a căror intersecție este intervalul 2 3. Dacă fișierul conține numerele 3 4 5 4 5 6 8 atunci pe ecran se va afișa multimea vida.

   a) Se cere să se proiecteze un algoritm eficient din punct de vedere al timpului de executare și al spațiului de memorie utilizat și să se realizeze o descriere de 3-4 rânduri a algoritmului ales, justificându-se eficiența acestuia. **(2 puncte)**

   b) Scrieți programul C/C++ corespunzător metodei descrise la punctul **a)**. **(8 puncte)**

Teza 17

SUBIECTUL I (20 de puncte)

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

1. Ştiind că variabila reală x aparţine reuniunii intervalelor [3, 7] şi (20, 30) care dintre expresiile scrise mai jos, NU are valoarea 1?
   a) !((x<3 ||x>7) && (x>=30 || x<=20))
   b) (x<=7 && x>=3) && (x>20 && x<30)
   c) (x<=7 && x>=3) || (x>20 && x<30)
   d) !(x<3 || x>7) || !(x>=30 ||x<=20)

2. Pentru definiţia alăturată a subprogramului P, stabiliţi ce se afişează la apelul P(4,1).

```
void P(int n, int i)
{  int j;
   if (i < n)
   {      P(n,i+1);
          for(j = 1; j<= i; j++)
          cout<<j+2;
   }
}
```

   a) 334345 b) 112123 c) 345343 d) 123121

3. Un arbore cu 6 noduri, numerotate de la 1 la 6, este reprezentat prin vectorul de tata = (x, y, 4, z, 0, 1). Ce valoare pot avea numerele x, y, z din acest vector, astfel încât arborele să aibă o singură frunză?
   a) 2, 3, 6 b) 1, 3, 4 c) 2, 3, 5 d) 2, 5, 3

4. Utilizând metoda backtracking, se generează, în ordine descrescătoare, toate numerele formate din 4 cifre pare şi care au prima cifră diferită de ultima cifră. Primele cinci numere generate sunt: 8886, 8884, 8882, 8880, 8866. Indicaţi următoarele două numere care sunt generate după numărul 4806.
   a) 4804, 4802 b) 4802, 4800 c) 4808, 4820 d) 4800, 4688

5. Un graf neorientat are 30 muchii şi fiecare nod al grafului are gradul un număr nenul. Doar şase dintre noduri au gradul un număr par, restul nodurilor având gradele numere impare. Care este numărul maxim de noduri pe care poate să le aibă graful?
   a) 36 b) 54 c) 30 d) 60

SUBIECTUL II — 40 de puncte)

1. Se consideră declarațiile de mai jos, în care variabila **p** memorează denumirea unui produs și data expirării lui. Variabila **azi** memorează data zilei curente. Scrieți o secvență de instrucțiuni C++ care afișează mesajul **produs expirat,** dacă la data zilei curente produsul a depășit data expirării sau mesajul **produsul nu este expirat,** în caz contrar. **(6 puncte)**

```
struct data
{int zi, luna, an;};
struct produs
{ char denumire[41];
data dexp ;};
produs p; data azi;
```

2. În secvența de instrucțiuni de mai jos variabilele **i** și **j** sunt de tip întreg, iar variabila **A** memorează un tablou bidimensional de tip caracter, cu 5 linii și 5 coloane, numerotate de la 0 la 4. Fără a utiliza alte variabile, scrieți una sau mai multe instrucțiuni care pot înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, tabloul memorat în variabila **A** să aibă elementele din figura de mai jos. **(6 puncte)**

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

```
A B C D E
E A B C D
D E A B C
C D E A B
B C D E A
```

3. 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.

```
citește n,y (y număr natural, y<10,
n număr natural nenul)
e←0
pentru i←1, n execută
  citește x (număr natural nenul)
  cx←x
  nr←0
  cât timp x≠0 execută
    dacă x%10=y
      atunci
        nr←nr+1
    x←[x/10]
  dacă nr%2=0 și nr>0
    atunci
    e←e+cx
scrie e
```

a) Scrieți ce se va afișa dacă se citesc, în această ordine, numerele 5, 3, 2334, 31, 1535, 330, 75. **(6 puncte)**

b) Scrieți un set de date de intrare care să determine afișarea valorii 1000. **(6 puncte)**

c) Scrieți programul C++ corespunzător algoritmului dat. **(10 puncte)**

d) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să se înlocuiască structura „pentru" cu alt tip de structură repetitivă. **(6 puncte)**

SUBIECTUL III (30 de puncte)

1. Se citește de la tastatură un șir care are cel mult 250 de caractere: litere și spații. Șirul este format din mai multe cuvinte, separate între ele printr-un spațiu. Scrieți un program C++ care modifică în memorie șirul de caractere, prin interschimbarea caracterelor din fiecare pereche formată din două caractere alăturate, unul fiind vocală și celălalt fiind consoană.
Exemplu: Dacă se citește șirul **exemple de probleme** se va afișa pe ecran șirul **xemplee ed porbelem**. **(10 puncte)**

2. Subprogramul **termen** are un singur parametru **n**, prin care primește un număr natural (**1 ≤ n ≤ 100**) și returnează termenul cu numărul de ordine **n** din șirul următor: 1, 2, **3**, 2, 1, 2, 3, 4, 5, **6**, 5, 4, 3, 2, 1, 2, 3, 4, 5, 6, 7, 8, **9**, 8, 7, 6, 5, 4,3, 2, 1 etc. Scrieți în limbajul C/C++ definiția completă a subprogramului **termen**.
Exemplu: Dacă **n = 10** atunci, după apel, subprogramul va returna valoarea **6.**
(10 puncte)

3. Fișierul **bac.txt** conține un șir format din cel puțin 2 și cel mult 10^9 numere naturale, fiecare număr având cel mult 9 cifre. Să se determine o **secvență de lungime maximă**, din șirul aflat în fișier, **cu proprietatea că ultima cifră a primului număr din secvență este egală cu ultima cifră a ultimului număr din secvență**. Scrieți un program C++ care să determine și să afișeze pe ecran lungimea acestei secvențe și numărul de ordine în fișier al primului număr din secvență, utilizând un algoritm eficient din punct de vedere al timpului de execuție. Dacă în șirul de numere din fișier nu există o astfel de secvență se va afișa mesajul **Nu există.** Dacă șirul conține mai multe astfel de secvențe, se va utiliza ultima secvență din fișier. Se consideră că primul număr din fișier are numărul de ordine egal cu 1.

Exemplu: Dacă fișierul `bac.txt` conține numerele 2 37 27 24 315 144 7 85 214 atunci se afișează pe ecran 6 4.

a) Descrieți în limbaj natural algoritmul proiectat și justificați eficiența acestuia. **(2 puncte)**

b) Scrieți programul C/C++ corespunzător algoritmului descris. **(8 puncte)**

Teza 18

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. Stabiliți care dintre următoarele expresii C++ are valoarea **1** dacă și numai dacă numărul întreg n este impar și divizibil cu 5.

a) !(n%2==0 && n%5==0)
b) !(n%2==0) && n%5!=0
c) !(n%2==0) && (n%5==0)
d) (n%2==1) || (n%5==0)

2. Ce valoare are funcția următoare la apelul f(5234)?

```
int f(int x)
{ if (x < 9)
    return x*x;
  else
    if(x%2==0)
        return x%10+f(x/10);
    else
        return (x%10)* f(x/10);
  }
```

a) 14 **b)** 85 **c)** 120 **d)** 26

3. Un arbore cu 7 noduri, numerotate de la 1 la 7 este reprezentat prin vectorul de „tați" (2, 0, 4, 2, 4, 7, 2). Stabiliți un nod ce poate fi ales ca rădăcină a arborelui, astfel încât numărul de niveluri sa fie maxim. Nivelul pe care este situat un nod este egal cu numărul de noduri parcurse de la rădăcină până la acesta.

a) 1 **b)** 4 **c)** 6 **d)** 2

4. Utilizând metoda backtracking, se generează toate posibilitățile de a forma buchete formate din cinci flori diferite, alese dintre florile {`lalea, narcisă, crin, crizantemă, trandafir, garoafă, frezie`}. Stabiliți câte buchete de flori se pot obține, încât fiecare buchet să nu conțină împreună `crin` și `trandafir`. Ordinea florilor alese într-un buchet nu contează.

a) 21 **b)** 24 **c)** 10 **d)** 11

5. Un graf neorientat notat G este graf complet și eulerian. Care este numărul maxim de muchii pe care îl poate avea graful G, știind că este format din cel mult 50 de noduri?

a) 1225 **b)** 1176 **c)** 1128 **d)** 1275

SUBIECTUL II (40 de puncte)

1. Se consideră declararea de mai jos, corespunzătoare unei fracții reprezentată prin două numere naturale, ce reprezintă numărătorul și numitorul fracției. Variabila **x** memorează datele pentru 20 de fracții. Scrieți o secvență de program C++ care determină și afișează numărul de fracții care au cel mai mic numitor. **(6 puncte)**

```
struct fractie
{int numitor,numarator;
};
fractie x[20];
```

2. În secvența de instrucțiuni de mai jos variabilele **i** și **j** sunt de tip întreg, iar variabila **A** memorează un tablou bidimensional de tip întreg, cu 5 linii și 5 coloane, numerotate de la 0 la 4. Fără a utiliza alte variabile, scrieți una sau mai multe instrucțiuni care pot înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, tabloul memorat în variabila **A** să aibă elementele din figura de mai jos. **(6 puncte)**

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

2	2	2	2	2
2	4	6	8	10
2	6	12	20	30
2	8	20	40	70
2	10	30	70	140

3. 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.

```
citește a (a număr natural nenul)
m←0
t←0
repetă
   x←a
   d←2
   nr←0
   cât timp a>1 execută
      dacă a%d=0
      atunci
         cât timp a%d=0 execută
            a=[a/d]
         nr ←nr+1
      d ←d+1
   dacă nr>=m
   atunci
      m←nr
      t←x
   citește a
până când a=0
scrie t
```

a) Scrieți ce se va afișa dacă se citesc în această ordine numerele 4, 7, 12, 35, 0. **(6 puncte)**

b) Scrieți un set de date de intrare care să determine afișarea unui număr prim. **(6 puncte)**

c) Scrieți programul C++ corespunzător algoritmului dat. **(10 puncte)**

d) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să se înlocuiască structura „repetă... până când" cu un alt tip de structură repetitivă. **(6 puncte)**

SUBIECTUL III (30 de puncte)

1. Se citește de la tastatură un cuvânt **c**, format din cel mult 20 de litere mici și un text cu cel mult 200 de caractere format din mai multe propozitii. O propoziție conține cuvinte separate prin spațiu și se încheie cu punct. Scrieți un program C++ care va afișa pe ecran o propoziție din text care conține cuvântul c de cele mai multe ori. De exemplu, dacă se citește cuvântul **avion** și textul **Din avion vezi un cer albastru și un alt avion. Pe aeroport aterizează un avion.** Se va afișa propoziția **Din avion vezi un cer albastru și un alt avion.** **(10 puncte)**

2. Subprogramul **perechi** primește prin parametrul **n** un număr natural (**2≤n≤50**) și prin parametrul **v** primește un tablou unidimensional, format din **n** numere naturale distincte, fiecare număr având cel mult **9** cifre. Subprogramul returnează numărul de perechi distincte formate din elemente din tabloul v, cu proprietatea că un număr este oglindit pentru celălalt număr. Scrieți în limbajul C++ definiția completă a subprogramului perechi.

   Exemplu: Dacă **n = 7** și **v = (41, 324, 14, 423, 82, 6, 28)** atunci, după apel, subprogramul va returna valoarea **3,** deoarece vectorul are 3 perechi de numere ce verifică proprietatea dată: (41,14); (324, 423); (82, 28). **(10 puncte)**

3. Fișierul **`bac.txt`** conține un șir format din cel mult 10 000 numere naturale separate prin spațiu, fiecare număr având cel mult 5 cifre. Prin alegerea cifrelor distincte din șir, care au modulul diferenței cifrelor egal cu 2 se pot obține mai multe numere. Scrieți un program C++ eficient ca timp de execuție și spațiu de memorie utilizat, care determină și afișează pe ecran cel mai mare număr format din toate cifrele numerelor din fișier, cu proprietatea că modulul diferenței cifrelor distincte este egal cu 2 sau afișează mesajul `Nu există`, dacă din șir nu se poate obține un asemenea număr.

   Exemplul 1: Dacă fișierul **`bac.txt`** conține numerele 1012 3367 9837 369 96 183, se pot obține mai multe numere care au modulul diferenței cifrelor distincte egal cu 2 : 20, 31, 311, 97, 79, 886, 9977 etc. Numărul care se va afișa este 33333111.

   Exemplul 2: Dacă fișierul **`bac.txt`** conține numerele 27 388 723 82, se va afișa mesajul Nu există.

   a) Descrieți în limbaj natural algoritmul proiectat și justificați eficiența acestuia. **(2 puncte)**

   b) Scrieți programul C/C++ corespunzător algoritmului descris. **(8 puncte)**

Teza 19

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. Stabiliți care dintre următoarele expresii C++ are valoarea 1, dacă și numai dacă numerele naturale nenule x și y au aceeași paritate (ambele sunt numere pare sau ambele sunt numere impare).

a) x+y %2==0 **b)**(x+y) %2==0 **c)** (x*y) %2==0 **d)** (x*y) %2==1

2. Se consideră funcțiile f1 și f2 definite mai jos. Stabiliți ce valoare are apelul f2(10, 23).

```
int f1(int n,int d)
{ if(d<=n/2)
if(n%d==0)
return d;
else
return f1(n,d+1);
else
return 0;
}
```

```
int f2(int a,int b)
{ if(a<b)
  if(f1(a,2)>0)
    return f2(a+1,b);
  else
    return 1+f2(a+1,b);
  else
    return 0;
}
```

a) 4 **b)** 5 **c)** 9 **d)** 10

3. Se consideră un arbore cu 5 noduri, notate de la 1 la 5, reprezentat prin matricea de adiacență de mai jos. Numim înălțime a unui arbore cu rădăcină, numărul de muchii al celui mai lung lanț elementar care are una dintre extremități în rădăcina arborelui. Care este nodul ales ca rădăcină astfel încât înălțimea arborelui să fie minimă?

```
0 1 1 0 0
1 0 0 1 0
1 0 0 0 1
0 1 0 0 0
0 0 1 0 0
```

a) 2 **b)** 1 **c)** 3 **d)** 5

4. Prin metoda backtracking, se generează în ordine lexicografică toate șirurile formate din 5 litere mici distincte, cu proprietatea că primul și ultimul caracter sunt vocale. Primele

trei șiruri obținute sunt în ordine: abcde, abcdi, abcdo. Știind că în alfabetul englez există 26 de litere mici, stabiliți câte șiruri care încep cu litera a se pot forma.

a) 16192 **b)** 3120 **c)** 358800 **d)** 48756

5. Fie un graf orientat G cu 6 noduri, reprezentat prin următoarele liste de adiacență: 1: (2, 3), 2: (6), 3: (2), 4: (5), 5:(4), 6: (4). Care dintre următoarele afirmații este falsă?

a) Graful G nu este tare conex.

b) Graful G are 5 componente tare conexe.

c) Graful G are 4 componente tare conexe.

d) Graful G devine tare conex dacă se adaugă cel puțin două arce noi.

SUBIECTUL II (40 de puncte)

1. În declararea următoare, variabila **c** memorează pentru fiecare 20 de cuvinte: cuvântul prin câmpul **cuv** și frecvența sa într-un text prin câmpul **frecv.** Scrieți o secvență de program C++ care determină și afișează lungimea maximă a cuvintelor memorate care au frecvența mai mare decât 2. **(6 puncte)**

```
struct cuvant
  { char cuv[30];int frecv;
  };
cuvant c[20];
```

2. În secvența de instrucțiuni de mai jos variabilele **i** și **j** sunt de tip întreg, iar variabila **A** memorează un tablou bidimensional de tip caracter, cu 4 linii și 4 coloane, numerotate de la 0 la 3. Fără a utiliza alte variabile, scrieți una sau mai multe instrucțiuni care pot înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, tabloul memorat în variabila **A** să aibă elementele din figura de mai jos. **(6 puncte)**

```
for(i=0;i<4;i++)
  for(j=0;j<4;j++)
    ...............
```

```
a a b b
A A B B
a a b b
A A B B
```

3. 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.

```
citește a,b (a≤ b, a, b numere naturale)
k←0; q←0
pentru x←1, a execută
    pentru y←1, b execută
        dacă y%x=0
        atunci
            k←k+y
        altfel
            q←q+y
scrie k," ",q
```

a) Scrieți valoarea care se va afișa pentru a = 3 și b = 7. **(6 puncte)**

b) Scrieți un set de valori pentru a și b astfel încât rezultatul afișat să fie k = 8 și q = 4. **(6 puncte)**

c) Scrieți programul C++ corespunzător algoritmului dat. **(10 puncte)**

d) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să se înlocuiască prima structură „pentru" cu alt tip de structură repetitivă. **(6 puncte)**

SUBIECTUL III (30 de puncte)

1. Se citesc de la tastatură două șiruri de caractere **s1** și **s2** formate din cel mult 40 de litere mici. Scrieți un program C++ care modifică în memorie cele două șiruri astfel: elimină vocalele din șirul **s1** și le inserează în șirul **s2** pe poziții pare, în ordine începând de la primul caracter. Să se afișeze pe ecran șirurile modificate.

Exemplu: Pentru șirurile `elicopter` și `cartile`, șirul **s1** devine `lcptr`, iar **s2** devine `eciaoretile`. **(10 puncte)**

2. Subprogramul **factor** are patru parametri: **n** prin care primește un număr natural (**1≤n≤50**), **a** prin care primește un tablou unidimensional format din n numere naturale nenule care au cel mult 3 cifre, **fp** prin care furnizează cel mai mic factor prim comun tuturor elementelor din tabloul a și **p** prin care furnizează puterea factorului fp pentru care numărul fp^p este divizor pentru fiecare element din tabloul a, cu p>0. Dacă elementele tabloului a nu au un factor prim comun atunci fp are valoarea –1 și p are valoarea 0. Scrieți în limbajul C++ definiția completă a subprogramului factor.

Exemplu: Pentru n = 4 și tabloul `a=(60,36,18,24)`, subprogramul factor va avea fp = 2, p = 1, iar pentru n = 4 și tabloul `a=(22,60,21,25)`, fp = `-1` și p = 0. **(10 puncte)**

3. Scrieți un program C++, eficient ca timp de execuție și spațiu de memorie utilizat, care afișează în fișierul **bac.txt**, în ordine strict crescătoare, pe câte o linie, toate palindroamele formate din 6 cifre impare. Primele 3 numere afișate în fișier sunt: 111111,113311,115511.

a) Descrieți în limbaj natural algoritmul proiectat și justificați eficiența acestuia. **(2 puncte)**

b) Scrieți programul C/C++ corespunzător algoritmului descris. **(8 puncte)**

Teza 20

SUBIECTUL I **(20 de puncte)**

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

1. Stabiliţi care dintre următoarele expresii C++ are valoarea 1 dacă şi numai dacă numerele naturale nenule x şi y sunt ambele divizibile cu 5.

a) (x+y)%5==0 **b)** (x*y)%5==0 **c)** (x/5+y/5)== (x+y)/5 **d)** x+y%5==0

2. Se consideră funcţia de mai jos.

```
void f(int n)
{ int i;
  if(n<5)
    cout<<"*";
  else
    if(n<10)
      cout<<"**";
    else
      for(i=1;i<=n/5;i++)
      {f(n/5);
       cout<<"*";
       }
}
```

Alegeţi 3 valori distincte ale numărului n, pentru care la apelul f(n) se vor afişa 18 caractere asterisc.

a) 10, 12, 14 **b)** 32, 33, 34 **c)** 24, 25, 26 **d)** 20, 21, 22

3. Se consideră un arbore cu 6 noduri, reprezentat prin vectorul de „taţi" t = (3, 3, 0, 3, 2, 2). Numim înălţime a unui arbore cu rădăcină, numărul de muchii al celui mai lung lanţ elementar care are una dintre extremităţi în rădăcina arborelui. Ce înălţime are arborele dat?

a) 3 **b)** 4 **c)** 1 **d)** 2

4. Prin metoda backtracking, se generează în ordine lexicografică toate modalităţile de scriere a unui număr natural ca sumă de numere naturale nenule şi pare. Pentru numărul 6 se obţin soluţiile: 2 + 2 + 2, 2 + 4, 6. A patra soluţie obţinută prin acelaşi algoritm pentru numărul 10 este:

a) 2+2+6 **b)** 2+8 **c)** 2+4+4 **d)** 2+2+2+4

5. Fie un graf neorientat G cu 6 noduri, notate de la 1 la 6, reprezentat prin următoarele liste de adiacență: 1: (2, 3), 2: (1, 3), 3: (1, 2, 4), 4: (3), 5: (6), 6: (5). Care dintre următoarele afirmații este adevărată pentru graful G?

a) Graful este eulerian.
b) Graful nu conține cicluri.
c) Graful este conex.
d) Graful nu are noduri izolate.

SUBIECTUL II (40 de puncte)

1. În declararea următoare, variabila **e** memorează pentru fiecare dintre 30 de elevi: numele elevului prin câmpul `nume`, prima limbă străină studiată prin câmpul `l1` și a doua limbă străină studiată prin câmpul `l2`. Scrieți o secvență de program C++ care determină și afișează numărul de elevi care studiază limba engleză ca prima sau a doua limbă străină. **(6 puncte)**

```
struct elev{
char l1[30],l2[30];char nume[40];};
elev e[30];
```

2. În secvența de instrucțiuni de mai jos variabilele **i** și **j** sunt de tip întreg, iar variabila **A** memorează un tablou bidimensional de tip întreg, cu 4 linii și 4 coloane, numerotate de la 0 la 3. Fără a utiliza alte variabile, scrieți una sau mai multe instrucțiuni care pot înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, tabloul memorat în variabila A să aibă elementele din figura de mai jos. **(6 puncte)**

```
for(i=0;i<4;i++)
  for(j=0;j<4;j++)
      ...............
```

1	2	3	**7**
1	**1**	**7**	4
1	**7**	**1**	4
7	2	3	**1**

3. 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.

```
citește a (a număr natural nenul)
m ←0
pentru d ←a, 2, –1 execută
    dacă a%d=0
        dacă d%2=1
        atunci
            m ←d
        altfel
            m ←2*d
scrie m
```

a) Scrieți ce se va afișa dacă se citește numărul 35. **(6 puncte)**

b) Scrieți un set de date de intrare care să determine afișarea numărului citit. **(6 puncte)**

c) Scrieți programul C++ corespunzător algoritmului dat. **(10 puncte)**

d) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să se înlocuiască structura „pentru" cu alt tip de structură repetitivă. **(6 puncte)**

SUBIECTUL III (30 de puncte)

1. Se citeşte de la tastatură un şir format din cel mult 20 de litere şi cifre. Scrieţi un program C++ care va determina şi va afişa pe ecran produsul numerelor din şir, ştiind că un număr are cel mult 3 cifre şi în şir există cel puţin cinci litere.
Exemplu: Dacă se citeşte şirul **abc123DE2c42e5** se va afişa numărul 51660 care reprezintă produsul numerelor 123, 2, 42 şi 5. **(10 puncte)**

2. Subprogramul **produs** are doi parametri: **n** prin care primeşte un număr natural (**1≤n≤100**) şi **x** prin care primeşte un tablou unidimensional format din n numere întregi şi returnează produsul maxim obţinut din două elemente situate pe poziţii distincte ale tabloului.
Exemplu: Pentru n = 6 şi tabloul a = (– 60, 36, – 8, – 2, 10, 5), subprogramul va returna 480. Scrieţi în limbajul C++ definiţia completă a subprogramului produs. **(10 puncte)**

3. Considerând un număr natural **x** ce conţine cel puţin trei cifre, după ce se elimină prima şi ultima cifră se obţine un alt număr denumit **număr interior** al numărului x. Fişierul **bac.in** conţine un şir având cel mult un milion de numere naturale care au cel puţin trei cifre şi cel mult 5 cifre, separate prin câte un spaţiu. Scrieţi un program C++ care afişează pe ecran în ordine descrescătoare, separate prin spaţiu, numerele interioare ale numerelor din fişier pentru care prima şi ultima lor cifră sunt egale sau afişează mesajul Nu există, dacă fişierul nu conţine niciun număr având prima şi ultima cifră egale. Utilizaţi un algoritm eficient ca timp de execuţie.
Exemplu: Dacă fişierul conţine şirul de numere 1151 234 2322 212 514 23122 atunci se vor afişa numerele 312 32 15 1.
a) Descrieţi în limbaj natural algoritmul proiectat şi justificaţi eficienţa acestuia. **(2 puncte)**
b) Scrieţi programul C/C++ corespunzător algoritmului descris. **(8 puncte)**

Teza 21

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. Se consideră o secvență de program C++, cu variabilele întregi i, j, n și s.

```
s=0;i=1;
do
{if(i+s<=n)
   {s=s+i;
    for(j=1;j<=i;j++)
    cout<<j<<" ";
    i++;
   }
else
s=n;
}
while(s<n);
```

Ce valoare are numărul n astfel încât la execuția acestei secvențe de program șirul numerelor afișate să conțină numărul 1 de patru ori?

a) 27 **b)** 16 **c)** 11 **d)** 4

2. Ce valoare are funcția următoare la apelul f(3,14)?

```
int f(int n,int m)
{if(n<m)
  if(n%2==0)
    return n+f(n+3,m);
  else
    return n*f(n+3,m);
else
return 1;
}
```

a) 162 **b)** 97 **c)** 369 **d)** 45

3. Se consideră un graf neorientat G cu 26 de noduri, notate cu toate literele mici din alfabetul englez. În graful G, oricare două noduri notate cu vocale sunt adiacente și oricare două noduri notate cu consoane sunt adiacente. Câte muchii are graful G?

a) 325 **b)** 256 **c)** 64 **d)** 220

4. Utilizând metoda backtracking, se generează în ordine crescătoare toate numerele naturale formate din patru cifre distincte și nenule cu proprietatea că au suma cifrelor un număr par. Primele trei numere obținute sunt: 1234, 1236, 1238. Ultimele trei numere generate sunt:

a) 9874,9875,9876 c) 9894,9896,9898
b) 8972,8974,8976 d) 9872,9874,9876

5. Se consideră un arbore cu rădăcină având **n** noduri. Fiecare nod de pe nivelul i are 2*i + 1 fii, cu excepția ultimului nod de pe ultimul nivel, obținut prin parcurgerea nivelului de la stânga la dreapta, care poate avea mai puțini fii. Știind că rădăcina se află pe nivelul 0, determinați valoarea lui **n** astfel încât pe nivelul 3 din arbore să se afle 11 noduri.

a) 22 b) 16 c) 25 d) 32

SUBIECTUL II (40 de puncte)

1. În declararea următoare, variabila **c** memorează pentru fiecare dintre cei 200 de candidați de la examenul de Bacalaureat: numele elevului prin câmpul `nume`, denumirea probei E_d alese prin câmpul `probaE_d` și nota obținută la această probă prin câmpul `nota`. Scrieți o secvență de program C++ care afișează numele elevilor care au obținut notă mai mare decât 9 la proba Informatica. **(6 puncte)**

```
struct candidat
{char nume[30],probaE_d[30];int nota;};
 candidat c[200];
```

2. Completați punctele de suspensie din secvența de instrucțiuni de mai jos, astfel încât la execuția secvenței să se afișeze pe ecran **info programinfo**. **(6 puncte)**

```
char s[30],t[20],x[20];
strcpy(s,"informatica");strcpy(t,"programare");
.......................
cout<<x<<" "<<t;
```

3. 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.

```
citește n (n număr natural nenul)
s ←0; d←2
pentru i ←1, n execută
  citește a
  dacă a%d=0 atunci
    s←s+d
  altfel
    s←s+1
    d←d+1
scrie s
```

a) Scrieți ce se va afișa dacă se citesc numerele 7 22 8 21 15 7 28 3. **(6 puncte)**

b) Scrieți două seturi distincte de date de intrare care să determine afișarea numărului 10. **(6 puncte)**

c) Scrieți programul C++ corespunzător algoritmului dat. **(10 puncte)**

d) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să se înlocuiască structura „pentru" cu alt tip de structură repetitivă. **(6 puncte)**

SUBIECTUL III (30 de puncte)

1. Scrieţi un program C++ care citeşte de la tastatură un număr natural **n** şi elementele, numere întregi, ale unui tablou bidimensional A, având **n** linii şi **n** coloane ($2 \le n \le 40$), determină valorile minime din coloanele tabloului A, care au elemente dispuse în ordine strict crescătoare şi afişează pe ecran media lor aritmetică sau afişează mesajul **Nu există coloane ordonate strict crescător**, dacă tabloul A nu are nicio coloană cu această proprietate.
Exemplu: Pentru n = 3 şi tabloul se va afişa media aritmetică a valorilor minime 2 şi 7 care este egală cu 4.5. **(10 puncte)**

2. Subprogramul **termeni** are 3 parametri:
– parametrul **n,** prin care primeşte un număr natural, având cel mult 8 cifre, n>1;
– parametrul **m,** prin care furnizează cel mai apropiat termen din şirul lui Fibonacci, mai mic decât n;
– parametrul **t,** prin care furnizează cel mai apropiat termen din şirul lui Fibonacci, mai mare decât n.
Scrieţi în limbajul C++ definiţia completă a subprogramului **termeni.**
Exemplu: Dacă n = 30 se obţin: m = 21, t = 34. **(10 puncte)**

3. Un şir de numere $x_1, x_2, \ldots x_k, \ldots x_m$ conţine o **secvenţă-munte** dacă are un element x_k $(1<k<m)$ şi există i şi p $(1 \le i<k<p \le m)$ cu proprietatea: $x_i<x_{i+1}<\ldots<x_k$ şi $x_k>x_{k+1}>\ldots>x_p$. Elementul x_k este denumit **vârf** şi se notează cu **V**. Fişierul **bac.txt** conţine pe prima linie un număr natural **n** ($4 \le n \le 10000$) şi pe doua linie un şir de n numere naturale nenule, având cel mult trei cifre. Scrieţi un program C++, eficient ca timp de execuţie şi spaţiu de memorie utilizat, care afişează pe ecran în ordine crescătoare, toate valorile V din toate secvenţele-munte distincte ce aparţin şirului de numere din fişier sau afişează mesajul **Nu există**, dacă nu există nicio secvenţă-munte în şir.
Exemplu: Pentru n = 14, şirul de numere 9, 8, **30**, 11, 14, 20, 1**25**, 24, 19, 34, **89**, 80, 56, 170 conţine trei secvenţe-munte cu vârfurile 30,125 şi 89 şi se va afişa pe ecran 30, 89, 125.
a) Descrieţi în limbaj natural algoritmul proiectat şi justificaţi eficienţa acestuia. **(2 puncte)**
b) Scrieţi programul C/C++ corespunzător algoritmului descris. **(8 puncte)**

Teza 22

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. Stabiliți ce valori au variabilele întregi a, b și c, astfel încât expresia `a/5+12.5-b/3*c` să aibă valoarea `2.5` în limbajul C++.

a) `a=16, b=14, c=4`
b) `a=2, b=10, c=3`
c) `a=11, b=14, c=3`
d) `a=2, b=16, c=3`

2. Se consideră un vector v = (6, –2, –10, 3, –15, –1, –8). Știind ca indicii elementelor din vector încep de la 0 stabiliți ce valoare are funcția următoare la apelul f(7).

```
int f(int n){
if(n>0)
 if((n-1)%2==1&&v[n-1]<0)
  return v[n-1]+f(n-1);
 else
 return f(n-1);
else
 return 0;
}
```

a) –11 **b)** –25 **c)** –3 **d)** 0

3. Se consideră un graf neorientat reprezentat prin matricea de adiacență alăturată și operațiile **AD (x, y)**: adăugarea în graf a muchiei de la nodul x la nodul y și **EL (x, y)**: eliminarea din graf a muchiei de la nodul x la nodul y. Stabiliți care dintre secvențele de operații următoare aplicate grafului îl transformă în arbore.

0	1	0	1	0	0
1	0	0	1	0	0
0	0	0	0	0	0
1	1	0	0	0	0
0	0	0	0	0	1
0	0	0	0	1	0

a) AD(2,3); AD(3,6); EL(5,6)
b) EL(1,2); EL(1,4); AD(3,6)
c) EL(2,4); AD(2,3); AD(3,5)
d) AD(2,5); AD(2,6); AD(2,3)

4. Se generează prin metoda backtracking șiruri de 4 caractere formate din caracterele `'+'` și `'-'` astfel încât orice șir generat nu conține două caractere `'-'` alăturate. Primele cinci

soluții obținute sunt ++++, +++-, ++-+, +-++, +-+-. Care sunt ultimele două șiruri obținute prin această generare?

a) --++, --+- b) -+++, -++- c) -++-, -+-+ d) ---+, ++-+

5. Un graf orientat format din 5 noduri este reprezentat prin listele de adiacență următoare: 1: (3, 5); 2: listă vidă; 3: (2); 4: (5); 5: listă vidă.
Care este numărul minim de arce ce trebuie adăugate grafului, astfel încât graful să fie tare conex?

a) 1 b) 2 c) 3 d) 4

SUBIECTUL II (40 de puncte)

1. Se consideră declarările de mai jos, în care tabloul **c** memorează punctajele obținute de 300 de participanți la un concurs sportiv, format din mai multe probe. Tabloul c este ordonat după câmpul **proba**. La fiecare probă participă mai mulți concurenți.

```
struct concurs{
char nume_concurent[25],proba[20];
int punctaj;}c[300];
```

Scrieți o secvență de instrucțiuni C++ care să afișeze pe ecran pentru fiecare probă din concurs, denumirea probei și media aritmetică a punctajelor obținute de concurenții participanți la proba respectivă. **(6 puncte)**

2. În secvența de instrucțiuni de mai jos, variabilele **i** și **j** sunt de tip întreg, iar variabila **A** memorează un tablou bidimensional de tip char, cu 5 linii și 5 coloane, numerotate de la 0 la 4. Fără a utiliza alte variabile, scrieți una sau mai multe instrucțiuni care pot înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, tabloul memorat în variabila A să aibă elementele din figura de mai jos. **(6 puncte)**

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

```
a b c d e
b a b c d
c b a b c
d c b a b
e d c b a
```

3. 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.

```
citește x (x număr natural nenul,
x>9)
k ←0
cât timp x≠0 execută
  citește y (y număr natural)
  dacă x/10%10= y%10 și y≠0
  atunci
    k ←k+1
  x ←y
scrie k
```

a) Scrieți ce se va afișa dacă se citesc numerele 135, 23, 122, 76, 44, 124, 0. **(6 puncte)**

b) Scrieți un set de date de intrare format din cel puțin patru numere care să determine afișarea numărului 0. **(6 puncte)**

c) Scrieți programul C++ corespunzător algoritmului dat. **(10 puncte)**

d) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să se înlocuiască structura „cât timp" cu alt tip de structură repetitivă. **(6 puncte)**

SUBIECTUL III **(30 de puncte)**

1. Se citește de la tastatură un text cu cel mult 500 de caractere (litere și spații), format din cuvinte separate între ele printr-un spațiu. Se citește un cuvânt **c,** format din cel mult 10 litere mici. Să se elimine din text toate aparițiile cuvântului **c** care sunt precedate de un cuvânt care începe cu o vocală. Să se afișeze textul dat, după modificarea sa în memorie sau să se afișeze mesajul **Textul nu s-a modificat**, în cazul în care nu s-a eliminat niciun cuvânt.

   Exemplu: Dacă se citește textul **Firma implementează proiecte de design interior proiecte de consolidare și proiecte de amenajări** și se citește cuvântul **proiecte**, se va afișa textul modificat: **Firma implementează de design interior de consolidare și proiecte de amenajări**. **(10 puncte)**

2. Subprogramul **prime** primește prin parametrul **n** un număr natural (**2≤n≤50**) și prin parametrul **a** primește un tablou unidimensional, format din **n** numere naturale nenule și distincte, fiecare număr având cel mult **9** cifre. Subprogramul returnează numărul de perechi distincte formate din elemente din tabloul v, prime între ele. Două numere naturale sunt prime între ele dacă au un singur divizor comun: numărul 1. Scrieți în limbajul C++ definiția completă a subprogramului prime.

   Exemplu: Dacă **n = 6** și **a = (42, 3, 14, 43, 8, 6)** atunci, după apel, subprogramul va returna valoarea **7,** deoarece vectorul are 7 perechi de numere ce verifică proprietatea dată: (42, 43); (3, 14); (3, 43); (3, 8); (14, 43); (43, 8); (43, 6). **(10 puncte)**

3. Fișierul **bac.txt** conține pe prima linie un număr natural **n,** care este multiplu de 7 (7≤n≤70000) și pe a doua linie un șir de n numere întregi nenule, având cel mult trei cifre.

Șirul de numere se împarte în secvențe de câte 7 numere denumite **benzi** și pentru fiecare bandă se determină cele mai mici două numere distincte **min1** și **min2**. Scrieți un program C++, eficient ca timp de execuție și spațiu de memorie utilizat, care afișează pe ecran, pentru fiecare dintre benzi, în ordine crescătoare cele două valori **min1** și **min2** sau afișează numărul 0, dacă banda nu conține două valori distincte **min1** și **min2**.

Exemplu: Pentru **n = 21**, șirul de numere **–3, 10, 1, 2, –5, 6, –5, 1, 1, 1, 1, 1, 1, 1, 10, 9, 1, 2, 3, 6, 7** se va afișa **–5 –3 0 1 2**.

a) Descrieți în limbaj natural algoritmul proiectat și justificați eficiența acestuia. **(2 puncte)**

b) Scrieți programul C/C++ corespunzător algoritmului descris. **(8 puncte)**

Teza 23

SUBIECTUL I **(20 de puncte)**

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

1. Stabiliţi ce valori au variabilele întregi **n** şi **x**, astfel încât la execuţia secvenţei de program următoare să se afişeze patru caractere * urmate de şase caractere #.

```
int i=1,n,x;
while(i*i<=n)
{cout<<'*'; i++;}
while(i<=n+x)
{cout<<'#';i++;}
```

a) n = 18, x = 2 **b)** n = 5, x = 6 **c)** n = 16, x = –6 **d)** n = 24, x = 5

2. Ce se va afişa la apelul **sir('f')** al funcţiei următoare?

```
void sir (char c){
char x;
if(c>'a')
    {x=c+1;
     cout<<x;
     sir(c-1);
     cout<<c;
}
}
```

a) gffeeddccb **b)** fedcbgfedc **c)** efcdgbefcd **d)** gfedcbcdef

3. Un arbore cu 48 noduri are următoarele proprietăţi:
– rădăcina se află pe nivelul 0;
– orice nivel notat cu un număr par, cu excepţia nivelului 0, are 10 noduri;
– orice nivel notat cu un număr impar are 9 noduri.

Numărul minim de frunze ale arborelui este:

a) 2 **b)** 4 **c)** 11 **d)** 16

4. Sorin are de cultivat în grădină patru tipuri de plante din cinci categorii de plante ornamentale pe care le are la dispoziție: azalee, magnolie, liliac, margarete și anemone. El dorește să nu cultive magnolia lângă liliac. Utilizând metoda backtracking, primele trei soluții generate sunt: (azalee, magnolie, margarete, anemone), (azalee, magnolie, anemone, margarete), (azalee, liliac, margarete, anemone). Care este a șasea soluție obținută?
   a) (azalee, margarete, liliac, magnolie)
   b) (azalee, anemone, liliac, margarete)
   c) (azalee, margarete, anemone, magnolie)
   d) (azalee, margarete, magnolie, anemone)

5. Se consideră un graf neorientat având 6 noduri, notate de la 1 la 6 și mulțimea muchiilor U={[1, 2], [1, 6], [2, 3], [2, 4], [2, 5], [2, 6], [3, 4], [3, 5], [4, 5], [5, 6]}. Prin eliminarea unui număr minim de muchii, graful devine eulerian. Care sunt muchiile ce trebuie eliminate?
   a) [2, 4] și [2, 5]
   b) [2, 5]
   c) [2, 6] și [4, 3]
   d) [2, 3] și [2, 6]

SUBIECTUL II (40 de puncte)

1. Se consideră declarările de mai jos, în care variabila **o** memorează date despre 200 de orașe: denumirea orașului, a țării căruia îi aparține orașul, numărul de locuitori și suprafața orașului (exprimată în număr de km^2).

```
struct oras{
char tara[30],nume_oras[20];
int nr_locuitori_oras; float suprafata_oras;}o[200];
```

Știind că densitatea populației reprezintă numărul de persoane care locuiesc pe un km^2, scrieți o secvență de instrucțiuni în limbajul C++ care să afișeze numele orașelor și țărilor lor, pentru toate orașele care au o densitate mai mare de 15 locuitori/km^2. **(6 puncte)**

2. Ce se va afișa după execuția secvenței de instrucțiuni următoare?

```
char s[20],t[10];
strcpy(s,"exercitii"); strcpy(t,"***");
cout<<strlen(s)+strlen(t)<<" ";
strcat(s,t);
cout<<strchr(s,'i');
```

(6 puncte)

3. 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.

citește a, b, k (numere naturale nenule, a≤b, k>1)
pentru x←a,b execută
  y ←0
  t←x
  cât timp t≠0 execută
    y←y*10+t%10
    t←[t/10]
  dacă y%k=0
    atunci
      scrie x," "

a) Scrieți ce se va afișa dacă se citesc numerele 12, 36, 3. **(6 puncte)**

b) Scrieți un set de date de intrare, cu proprietatea b –a = 2, pentru care se afișează trei numere. **(6 puncte)**

c) Scrieți programul C++ corespunzător algoritmului dat. **(10 puncte)**

d) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să se înlocuiască structura „pentru" cu alt tip de structură repetitivă. **(6 puncte)**

SUBIECTUL III (30 de puncte)

1. Scrieți un program C++ care citește de la tastatură un număr natural **n** și elementele, numere naturale, ale unui tablou bidimensional A, având **n** linii și **n** coloane ($2 \le n \le 50$) și înlocuiește toate elementele care sunt numere pare, formate din cel mult 3 cifre, cu cel mai mare element situat pe diagonala principală sau diagonala secundară a matricei, apoi afișează matricea transformată.

Exemplu: Pentru n = 3 și tabloul A = $\begin{matrix} 2 & 1000 & 7 \\ 40 & 8 & 27 \\ 5 & 2 & 9 \end{matrix}$ se va afișa matricea transformată

A = $\begin{matrix} 9 & 1000 & 7 \\ 9 & 9 & 27 \\ 5 & 9 & 9 \end{matrix}$ **(10 puncte)**

2. Subprogramul **cifre** primește prin parametrul **n** un număr natural nenul, format din cel mult 9 cifre, elimină toate cifrele pare ale numărului n și furnizează prin parametrul **n** numărul modificat și prin parametrul **k** un număr format din cifrele pare distincte ale numărului n, în ordinea scrierii lor. Parametrul k este egal cu –1 dacă numărul n nu are cifre pare.

Exemplu: Pentru numărul n = 12267488, se obține n = 17 și k = 2648, iar pentru n = 1331, se obține n = 1331 și k = –1. Scrieți în limbajul C++ definiția completă a subprogramului **cifre**. **(10 puncte)**

3. Fișierul **bac.in** conține un șir format din cel mult 10^9 litere ce aparțin mulțimii M = {a, A, b, B, c, C, d, D e, E}. Scrieți un program C++, eficient ca timp de execuție, ce afișează pe ecran prin litere mici, caracterele ce apar de un număr impar de ori în șirul dat, în ordinea enumerării lor în mulțimea M, fără a face distincție între litere mari și litere mici sau afișează mesajul Nu există, dacă șirul dat nu conține niciun caracter cu proprietatea menționată.

Dacă fișierul bac.txt conține caracterele `cbcCdEaAdBacCDEaacc` atunci pe ecran se va afișa `aaaaaccccccddd`. Dacă fișierul bac.txt conține caracterele `cbcBaa`, pe ecran se va afișa mesajul Nu există.

a) Descrieți în limbaj natural algoritmul proiectat și justificați eficiența acestuia. **(2 puncte)**

b) Scrieți programul C/C++ corespunzător algoritmului descris. **(8 puncte)**

Teza 24

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. Se consideră două numere întregi x și y și expresia `e=(x+10)>y?x:y`. Pentru ce valori ale lui x și y, expresia este egală cu y?

a) x = 3, y = –5 **b)** x = 10, y = 20 **c)** x = 20, y = 10 **d)** x = –5, y = 3

2. Se consideră șirul de caractere s = "abddcdebc". Ce valoare are funcția următoare la apelul sir(s, 0)?

```
int sir(char s[],int i){
if(i<strlen(s)/2)
  if(s[i]==s[strlen(s)-i-1])
   return 1+sir(s,i+1);
  else
   return sir(s,i+1);
else
 return 0;
}
```

a) 0 **b)** 1 **c)** 2 **d)** 3

3. Se consideră un arbore ce conține **n** muchii ($5 \le n \le 10$). Nodurile arborelui sunt etichetate cu numere prime distincte, consecutive în mulțimea numerelor prime, începând de la 2. În ce interval sunt situate aceste numere prime?

a) [2,10] **b)** [2,31] **c)** [11,29] **d)** [29,79]

4. Având la dispoziție suma S egală cu 100 de lei și șirul de bancnote exprimate **în lei:** (10, 20, 30, 40), se poate exprima suma S cu bancnotele date în mai multe moduri.
Utilizând metoda backtracking, o modalitate de exprimare a sumei S are forma x = (x_1, x_2, x_3, x_4), unde x_1 reprezintă numărul de bancnote de 10 lei, x_2 reprezintă numărul de bancnote de 20 lei, x_3 reprezintă numărul de bancnote de 30 lei și x_4 reprezintă numărul de bancnote de 40 lei.
Exemplu: Soluția (2, 2, 0, 1) exprimă suma 100 = 2*10 + 2*20 + 0*30 + 1*40. Dacă primele cinci soluții obținute sunt: (10, 0, 0, 0), (8, 1, 0, 0), (7, 0, 1, 0), (6, 2, 0, 0), (6, 0, 0, 1), atunci următoarele două soluții generate în ordine imediat după soluția (4, 3, 0, 0) sunt:

a) (4, 1, 0, 1) și (2, 4, 0, 0)
b) (3, 0, 1, 1) și (2, 4, 0, 0)
c) (4, 1, 0, 1) și (1, 1, 1, 1)
d) (4, 1, 0, 1) și (3, 0, 1, 1)

5. Un graf neorientat conține 5 noduri, notate de la 1 la 5. Oricare două noduri notate cu numere prime între ele sunt adiacente. Care dintre următoarele afirmații nu este adevărată?
   a) Graful este conex.
   b) Graful este complet.
   c) Graful este hamiltonian.
   d) Graful nu este eulerian.

SUBIECTUL II (40 de puncte)

1. Se consideră declarările de mai jos, în care tabloul **f** memorează date despre filmele difuzate la un cinematograf. Pentru fiecare film se memorează: **numele filmului**, **genul** (istoric, actiune, comedie, SF etc.), data difuzării filmului denumită **difuzare** și suma încasată din vânzarea biletelor la data difuzării denumită **suma**.

```
struct data{
int zi,luna,an;};
struct film{
char nume_film[40],gen_film[20];
data difuzare;
int suma;}f[1000];
```

Scrieți o secvență de instrucțiuni C++ care să determine și să afișeze pe ecran suma totală încasată de cinematograf, pentru toate filmele ce aparțin genurilor **comedie** și **istoric**, difuzate în luna martie, anul 2018. **(6 puncte)**

2. În secvența de instrucțiuni de mai jos variabilele **i** și **j** sunt de tip întreg, iar variabila **A** memorează un tablou bidimensional de tip întreg, cu 5 linii și 5 coloane, numerotate de la 0 la 4. Fără a utiliza alte variabile, scrieți una sau mai multe instrucțiuni care pot înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, tabloul memorat în variabila A să aibă elementele din figura de mai jos. **(6 puncte)**

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

1	2	3	4	-2
3	5	7	2	-2
8	12	9	0	-2
20	21	9	-2	-2
41	30	7	-4	-2

3. 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.

```
citește a, b (a, b numere naturale
nenule, a>b)
n ←0
repetă
citește x (x număr natural nenul)
t←0
y←x
repetă
    dacă x%10=a%b%10
    atunci
        t ←t+1
    x←[x/10]
până când x=0
dacă t≥2
atunci
    n←n+1
până când y=0
scrie n
```

a) Scrieți ce se va afișa dacă se citesc numerele 40 16 10 2880 18 80838 22 19818 0. **(6 puncte)**

b) Scrieți un set de date de intrare care să determine afișarea numărului 0. **(6 puncte)**

c) Scrieți programul C++ corespunzător algoritmului dat. **(10 puncte)**

d) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să se înlocuiască prima structură repetitivă cu test final din algoritm cu o structură repetitivă cu test inițial. **(6 puncte)**

SUBIECTUL III **(30 de puncte)**

1. Se citește de la tastatură un text, având cel mult 400 de caractere care sunt litere mici și spații, format din mai multe cuvinte separate printr-un singur spațiu. Scrieți un program C++ care modifică în memorie textul dat astfel: literele oricărui cuvânt din text care are proprietate palindromică se transformă în litere mari. Textul modificat se va afișa pe ecran. **Exemplu:** Dacă textul este: **tot ce este rar cojoc si radar** atunci se va afișa **TOT ce este RAR COJOC si RADAR.** **(10 puncte)**

2. Subprogramul **maxim** primește prin parametri **n** și **m** două numere naturale ($2 \le n, m \le 30$) și prin parametrul **T** un tablou bidimensional format din n linii și m coloane. Elementele tabloului sunt numere naturale. Subprogramul returnează cel mai mare număr par format numai din cifre pare, din prima linie a tabloului T care începe cu un număr impar sau returnează valoarea –1 dacă nu există un astfel de număr în tablou. Scrieți în limbajul C/C++ definiția completă a subprogramului **maxim.** **(10 puncte)**

3. Fișierul **bac.in** conține un șir format din cel mult 100 000 numere naturale separate prin spațiu, care au exact 3 cifre. Scrieți un program C++, eficient ca timp de execuție și spațiu

5. Un graf neorientat conține 5 noduri, notate de la 1 la 5. Oricare două noduri notate cu numere prime între ele sunt adiacente. Care dintre următoarele afirmații nu este adevărată?
   a) Graful este conex.
   b) Graful este complet.
   c) Graful este hamiltonian.
   d) Graful nu este eulerian.

SUBIECTUL II (40 de puncte)

1. Se consideră declarările de mai jos, în care tabloul **f** memorează date despre filmele difuzate la un cinematograf. Pentru fiecare film se memorează: **numele filmului**, **genul** (istoric, actiune, comedie, SF etc.), data difuzării filmului denumită **difuzare** și suma încasată din vânzarea biletelor la data difuzării denumită **suma**.

```
struct data{
int zi,luna,an;};
struct film{
char nume_film[40],gen_film[20];
data difuzare;
int suma;}f[1000];
```

Scrieți o secvență de instrucțiuni C++ care să determine și să afișeze pe ecran suma totală încasată de cinematograf, pentru toate filmele ce aparțin genurilor **comedie** și **istoric**, difuzate în luna martie, anul 2018. **(6 puncte)**

2. În secvența de instrucțiuni de mai jos variabilele **i** și **j** sunt de tip întreg, iar variabila **A** memorează un tablou bidimensional de tip întreg, cu 5 linii și 5 coloane, numerotate de la 0 la 4. Fără a utiliza alte variabile, scrieți una sau mai multe instrucțiuni care pot înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, tabloul memorat în variabila A să aibă elementele din figura de mai jos. **(6 puncte)**

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

1	2	3	4	-2
3	5	7	2	-2
8	12	9	0	-2
20	21	9	-2	-2
41	30	7	-4	-2

3. 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.

citește a, b (a, b numere naturale nenule, a>b)
n ←0
repetă
  citește x (x număr natural nenul)
  t←0
  y←x
  repetă
    dacă x%10=a%b%10
      atunci
        t ←t+1
    x←[x/10]
  până când x=0
  dacă t≥2
    atunci
      n←n+1

de memorie utilizat, care determină și afișează pe ecran **lungimea maximă a unei secvențe** din șirul aflat în fișier, cu proprietatea că toate numerele secvenței încep cu aceeași cifră și **numărul de secvențe de lungime maximă** de acest tip din fișier.

Exemplu: Dacă fișierul conține numerele 103 879 867 809 876 234 243 367 343 342 885 801 786 754 724 721, atunci se va afișa **4 2**.

a) Descrieți în limbaj natural algoritmul proiectat și justificați eficiența acestuia. **(2 puncte)**

b) Scrieți programul C/C++ corespunzător algoritmului descris. **(8 puncte)**

Teza 25

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 `x, y, a, b` memorează numere naturale. Care dintre următoarele expresii este adevarată dacă și numai dacă `[x,y]⊂[a,b]`?

a) `a<=x||b<=y` **b)** `a<=x&&y<=b` **c)** `!(a<=x&&y<=b)` **d)** `x<=a&&b<=y`

2. Se consideră graful orientat cu 8 noduri numerotate de la 1 la 8 și cu arcele: (1,3),(3,2),(2,1),(3,4),(4,7),(7,8),(8,5),(5,6),(6,4). În câte moduri putem adăuga un singur arc, astfel încât graful să devină tare conex?

a) 12 **b)** 8 **c)** 1 **d)** 15

3. La examenul de Bacalaureat, un elev trebuie să genereze prin metoda backtracking permutările mulțimii {A, B, C, D, E}, în care vocalele să fie precedate de consoane. Dacă primele trei permutări generate sunt BCDAE, BCDEA, CBDAE, la a câta permutare va fi obținut șirul DCBAE?

a) 12 **b)** 10 **c)** 11 **d)** 9

4. Un graf neorientat se numește pădure dacă toate componentele sale conexe sunt arbori. Fiind dat un graf neorientat cu 10 noduri și două componente conexe grafuri complete, una cu 21 de muchii și una cu 3 muchii, determinați numărul minim de muchii care trebuie eliminate din graf astfel încât graful să devină pădure.

a) 10 **b)** 8 **c)** 16 **d)** 17

5. Se consideră subprogramul f definit mai jos!

```
void f(int x, char c)
{   int i;
    if(x>1)
    {   cout<<"*";
        f(x-1,c+1);
        cout<<c<<"#";;
    }
    else
     cout<<c;
}
```

Ce se va afișa în urma apelului f(4,'a')?

a) *#*#*#abcd **b)** *a#*b#*c#*d **c)** ***dc#b#a# **d)** *#d*#c*#b*#a

SUBIECTUL II (40 de puncte)

1. Se consideră algoritmul alăturat, descris în pseudocod.

```
citește n (n număr natural)
x←7
pentru i←1, n execută
  j←2
  cât timp j≤[x/j] și x%j0 execută
      j←j+1
  ■
  dacă j*j>x atunci
      scrie x,' '
  ■
  x←x+10
■
```

a) Scrieți ce se va afișa dacă pentru n se citește valoarea 5. **(6 puncte)**

b) Scrieți cea mai mare valoare care poate fi citită pentru n astfel încât să se afișeze 5 valori. **(6 puncte)**

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

d) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să să înlocuiți structura „cât timp" cu o structură repetitivă cu test final. **(6 puncte)**

2. Se consideră secvența de program C++ următoare:

```
char voc[]="aeiou",s[100];
int i;
strcpy(s,"*galbbbleinn*asaa*appaarei*astfel*creuzet*");
for(i=0;i<strlen(s)-1;i++)
    {...}
cout<<s;
```

Completați punctele de suspensie cu o secvență de instrucțiuni C++, astfel încât să se afișeze șirul

***galen*asa*apare*asel*cezet*.** **(6 puncte)**

3. Se consideră declarările de mai jos, în care tabloul **t** memorează laturile pentru **n** triunghiuri. ($1 \le n < 100$) și $x \le y$.

```
struct triunghi{
float l1,l2,l3;}
t[100];
float x,y;
int n,i;
```

Scrieți o secvență de instrucțiuni C++ care să afișeze pe ecran laturile triunghiurilor pentru care aria aparține intervalului [x,y], separând triunghiurile prin caracterul *. **(6 puncte)**

SUBIECTUL III (30 de puncte)

1. Scrieți un program C/C++ care citește de la tastatură un număr natural N ($N \in [2,10]$), construiește în memorie și afișează un tablou bidimensional cu N linii și 2*N-1 coloane, numere naturale din mulțimea {1, 2}, astfel încât valoarea elementelor de pe semidiagonalele care încep cu primul element din stânga sus respectiv primul element din dreapta sus să fie

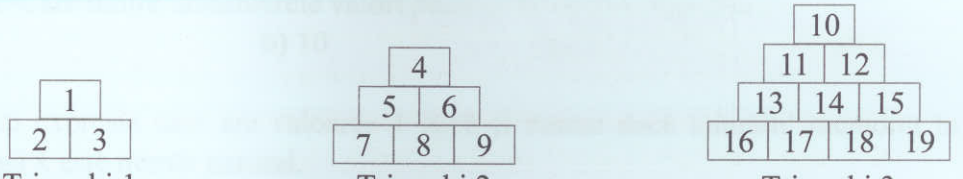
egale cu 1, celelalte elemente fiind egale cu 2. Elementele matricei vor fi afișate pe ecran linie cu linie separate printr-un spațiu. **(10 puncte)**

Exemplu: Dacă `N=3` atunci tabloul este:

```
1 2 2 2 1
2 1 2 1 2
2 2 1 2 2
```

2. Un tablou unidimensional se numește **k-palindrom** dacă după efectuarea a `k` permutări circulare cu o poziție spre stânga acesta devine palindrom (considerăm că un vector este palindrom dacă vectorul parcurs de la stânga la dreapta coincide cu vectorul parcurs de la dreapta la stânga). Scrieți definiția completă a unui subprogram **`kpal`** cu trei parametri: `v` un tablou unidimensional cu cel mult 100 de elemente numere întregi, `n` un număr natural (`n≤100`), reprezentând numărul de elemente din vector și `k` un număr natural (`1≤k<n`), care returnează valoarea 1 dacă vectorul este k-palindrom sau valoarea 0 în caz contrar.
Exemplu: Dacă n=5, v=(2, 2, 4, 5, 4) atunci f(v,n,1) va returna valoarea 1. **(10 puncte)**

3. Cu șirul crescător al numerelor naturale de la 1 la 10^6, se construiesc triunghiuri de numere ca în imaginea de mai jos (ultimul triunghi poate fi incomplet).



Triunghi 1 Triunghi 2 Triunghi 3

Se citește de la tastatură un număr k, $k < 10^6$. Să se scrie în fișierul text **BAC.TXT** pe prima linie, separate prin câte un spațiu, valorile T L C, reprezentând: T = numărul triunghiului care conține valoarea k, L = linia pe care se află valoarea k (numerotată de sus în jos, vârf = linia 1), C = coloana acestuia (numerotată de la stânga la dreapta). Proiectați un algoritm eficient din punct de vedere al timpului de executare și al spațiului de memorie utilizat.

Exemplu: Dacă se citește de la tastatură numărul k=17 atunci în fișier se vor afișa:

3 4 2

a) Realizați o descriere de 3-4 rânduri a algoritmului ales justificând eficiența acestuia. **(2 puncte)**

b) Scrieți programul C/C++ corespunzător metodei descrise la punctul **a)**. **(8 puncte)**