

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Utilizând metoda backtracking se generează în ordine lexicografică cuvintele de câte patru litere din mulțimea  $A=\{a,b,c,d,e\}$ , cuvinte care nu conțin două vocale alăturate. Primele opt cuvinte generate sunt, în ordine: **abab**, **abac**, **abad**, **abba**, **abbb**, **abbc**, **abbd**, **abbe**. Câte dintre cuvintele generate încep cu litera **b** și se termină cu litera **e**? **(4p.)**
- a. 9                                      b. 15                                      c. 12                                      d. 20

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Pentru definiția de mai jos a subprogramului **f**, ce se afișează ca urmare a apelului **f(121,1);**? **(6p.)**
- |  |   |
|--|---|
| <pre>//C void f(long n, int i) { if(n!=0)   if(n%3&gt;0)     { printf("%d",i); f(n/3,i+1); } }</pre> | <pre>//C++ void f(long n, int i) { if(n!=0)   if(n%3&gt;0)     { cout&lt;&lt;i; f(n/3,i+1); } }</pre> |
|--|---|
3. Fișierul text **bac.txt** conține, pe o singură linie, cel mult 1000 de numere naturale nenule cu cel mult 4 cifre fiecare, numerele fiind separate prin câte un spațiu. Scrieți un program C/C++ care citește de la tastatură un număr natural nenul **n** ( $n \leq 999$ ) și numerele din fișierul **bac.txt** și care afișează pe ecran, separate prin câte un spațiu, toate numerele din fișier care sunt divizibile cu **n**. Dacă fișierul nu conține niciun astfel de număr, atunci se va afișa pe ecran mesajul **NU EXISTA**.  
**Exemplu:** dacă fișierul **bac.txt** conține numerele: 3 100 40 70 25 5 80 6 3798, pentru **n=10** atunci pe ecran se va afișa: 100 40 70 80 **(10p.)**
4. Subprogramul **sub**, cu trei parametri, primește prin intermediul parametrilor:
- **v** un tablou unidimensional cu cel mult 100 de componente ce memorează numere întregi de cel mult 4 cifre fiecare;
  - **n** un număr natural nenul mai mic sau egal cu 100 ce reprezintă numărul efectiv de componente ale tabloului primit prin intermediul parametrului **v**;
  - **a** un număr întreg cu cel mult 4 cifre.
- Subprogramul **sub** returnează numărul componentelor tabloului primit prin intermediul parametrului **v** ale căror valori sunt strict mai mici decât valoarea parametrului **a**.  
**Exemplu:** pentru valorile **n=5**, **v=(1,21,9,21,403)**, **a=20** ale parametrilor, în urma apelului, subprogramul **sub** va returna valoarea 2.
- a) Scrieți definiția completă a subprogramului **sub**. **(4p.)**
- b) Să se scrie un program C/C++ care să citească de la tastatură un număr natural nenul **n** ( $n \leq 100$ ) și **n** numere întregi, fiecare având cel mult 4 cifre, și care, folosind apeluri utile ale subprogramului **sub**, să afișeze pe ecran mesajul **DA** dacă oricare două dintre cele **n** numere întregi citite sunt distincte două câte două, sau mesajul **NU** în caz contrar.  
**Exemplu:** pentru **n=6** și cele **n** numere citite de la tastatură: 47 183 69 8 134 -56 se va afișa pe ecran mesajul **DA** **(6p.)**

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Utilizând metoda backtracking se generează în ordine lexicografică cuvintele de câte patru litere din mulțimea  $A = \{a, b, c, d, e\}$ , cuvinte care nu conțin două vocale alăturate. Primele opt cuvinte generate sunt, în ordine: **abab**, **abac**, **abad**, **abba**, **abbb**, **abbc**, **abbd**, **abbe**. Care este ultimul cuvânt generat? **(4p.)**
- a. edcb                      b. eeee                      c. edde                      d. eded

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Pentru definiția de mai jos a subprogramului **f**, ce se afișează ca urmare a apelului **f(12345);?** **(6p.)**

<pre>//C void f(long n) { printf("%d",n%10);   if(n!=0)     { f(n/100); printf("%d",n%10); } }</pre>	<pre>//C++ void f(long n) { cout&lt;&lt;n%10;   if(n!=0)     { f(n/100); cout&lt;&lt;n%10; } }</pre>
--	--

3. Fișierul text **NR.TXT** conține pe o singură linie, separate prin câte un spațiu, cel mult 100 de numere **întregi**, fiecare număr având cel mult 4 cifre. Scrieți un program C/C++ care citește numerele din fișierul **NR.TXT** și afișează pe ecran, separate prin câte un spațiu, în ordine crescătoare, toate numerele **naturale nenule** din fișier. Dacă nu există astfel de numere se va afișa pe ecran mesajul **NU EXISTA**.

**Exemplu:** dacă fișierul **NR.TXT** conține numerele: -3 -10 0 7 -5 7 51 -800 6 3798, atunci pe ecran se va afișa: 6 7 7 51 3798 **(10p.)**

4. Un număr **n** se numește **extraprim** dacă atât el, cât și orice număr obținut prin permutarea cifrelor lui **n**, sunt numere prime. De exemplu, numărul 113 este un număr **extraprim** deoarece 113, 311, 131 sunt numere prime.

**a)** Scrieți definiția completă a unui subprogram **f**, cu un parametru, subprogram care:

- primește prin intermediul parametrului **a** un număr natural cu cel mult 3 cifre ( $a > 1$ )
- returnează suma tuturor exponenților din descompunerea în factori primi a valorii parametrului **a**.

**Exemplu:** pentru  $a=90$  subprogramul va returna valoarea 4, deoarece  $a=2 \cdot 3^2 \cdot 5$  și  $1+2+1=4$ . **(4p.)**

**b)** Scrieți un program C/C++ care citește de la tastatură un număr natural **n**,  $2 \leq n \leq 999$  și, folosind apeluri utile ale subprogramului **f**, verifică dacă **n** este un număr **extraprim**. În caz afirmativ, programul afișează pe ecran mesajul **DA**, în caz contrar afișând mesajul **NU**. **(6p.)**

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Utilizând metoda backtracking se generează în ordine lexicografică cuvintele de câte patru litere din mulțimea  $A = \{a, b, c, d, e\}$ , cuvinte care nu conțin două vocale alăturate. Primele opt cuvinte generate sunt, în ordine: **abab**, **abac**, **abad**, **abba**, **abbb**, **abbc**, **abbd**, **abbe**. Care este penultimul cuvânt generat? **(4p.)**
- a. edec                      b. eded                      c. edde                      d. edcb

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Pentru definiția alăturată a subprogramului **f**, ce se afișează ca urmare a apelului **f(26);**? **(6p.)**
- ```
void f (int x)
{
    if(x>0)
    if(x%4==0)
        { cout<<'x'; | printf("%c",'x');
          f(x-1); }
    else
        { f(x/3);
          cout<<'y'; | printf("%c",'y');
        }
}
```
3. Fișierului text **NR.TXT** conține pe o singură linie, separate prin câte un singur spațiu, cel mult 100 de numere naturale, fiecare număr având cel mult 4 cifre. Scrieți un program C/C++ care citește toate numerele din fișierul **NR.TXT** și afișează pe ecran, separate prin câte un spațiu, în ordine crescătoare, toate numerele din fișier care au cel puțin 3 cifre. Dacă fișierul nu conține astfel de numere se va afișa pe ecran mesajul **NU EXISTA**. **(10p.)**
4. Subprogramul **cif**, cu doi parametri, primește prin intermediul parametrului **a** un număr natural cu cel mult 8 cifre și prin intermediul parametrului **b** o cifră; subprogramul returnează numărul de apariții ale cifrei **b** în scrierea numărului **a**.  
**Exemplu:** pentru **a=125854** și **b=5**, subprogramul va returna valoarea 2.
- a) Scrieți definiția completă a subprogramului **cif**. **(4p.)**
- b) Scrieți un program C/C++ care citește de la tastatură un număr natural **n** cu **exact** 8 cifre și care determină și afișează pe ecran, folosind apeluri utile ale subprogramului **cif**, cel mai mare număr palindrom ce poate fi obținut prin rearanjarea tuturor cifrelor numărului **n**. Dacă nu se poate obține un palindrom din toate cifrele numărului **n**, programul va afișa pe ecran numărul 0. Un număr natural este palindrom dacă este egal cu numărul obținut prin scrierea cifrelor sale în ordine inversă.  
**Exemplu:** dacă **n=21523531** atunci se va afișa pe ecran numărul 53211235, iar dacă **n=12272351** atunci se va afișa pe ecran numărul 0. **(6p.)**

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Utilizând metoda backtracking se generează în ordine lexicografică cuvintele de câte patru litere din mulțimea  $A = \{a, b, c, d, e\}$ , cuvinte care nu conțin două vocale alăturate. Primele opt cuvinte generate sunt, în ordine: **abab**, **abac**, **abad**, **abba**, **abbb**, **abbc**, **abbd**, **abbe**. Care este antepenultimul cuvânt generat? **(4p.)**
- a. **edde**                      b. **eddb**                      c. **edeb**                      d. **edcb**

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Pentru definiția alăturată a subprogramului **f**, ce se afișează ca urmare a apelului **f(15,2);**? **(6p.)**
- ```
void f (int n, int x)
{ if(x>n)
  cout<<0; | printf("%d",0);
  else
  if(x%4<=1) f(n,x+1);
  else
  { f(n,x+3);
    cout<<1; | printf("%d",1);
  }
}
```
3. Fișierul text **NR.TXT** conține pe o singură linie, separate prin câte un singur spațiu, cel mult 100 de numere naturale, fiecare număr având cel mult 4 cifre. Scrieți un program C/C++ care citește numerele din fișierul **NR.TXT** și afișează pe ecran, separate prin câte un spațiu, în ordine descrescătoare, toate numerele din fișier care au cel mult 2 cifre. Dacă fișierul nu conține astfel de numere se va afișa pe ecran mesajul **NU EXISTA**. **(10p.)**
4. Subprogramul **cif**, cu doi parametri, primește prin intermediul parametrului **a** un număr natural cu cel mult 8 cifre și prin intermediul parametrului **b** o cifră; subprogramul returnează numărul de apariții ale cifrei **b** în scrierea numărului **a**.  
**Exemplu:** pentru **a=125854** și **b=5**, subprogramul va returna valoarea 2.
- a) Scrieți definiția completă a subprogramului **cif**. **(4p.)**
- b) Scrieți un program C/C++ care citește de la tastatură un număr natural **n** cu **exact** 8 cifre, fiecare cifră fiind nenulă, și care determină și afișează pe ecran, folosind apeluri utile ale subprogramului **cif**, cel mai mic număr palindrom ce poate fi obținut prin rearanjarea tuturor cifrelor numărului **n**. Dacă nu se poate obține un palindrom din toate cifrele numărului **n**, programul va afișa pe ecran numărul 0. Un număr natural este palindrom dacă este egal cu numărul obținut prin scrierea cifrelor sale în ordine inversă.  
**Exemplu:** dacă **n=21523531** atunci se va afișa pe ecran numărul 12355321, iar dacă **n=12272351** atunci se va afișa pe ecran numărul 0. **(6p.)**

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Folosind modelul combinărilor se generează numerele naturale cu câte trei cifre distincte din mulțimea  $\{1, 2, 3, 7\}$ , numere cu cifrele în ordine strict crescătoare, obținându-se, în ordine: 123, 127, 137, 237. Dacă se utilizează exact aceeași metodă pentru a genera numerele naturale cu patru cifre distincte din mulțimea  $\{1, 2, 3, 4, 5, 6, 7, 8\}$ , câte dintre numerele generate au prima cifră 2 și ultima cifră 7? (4p.)

a. 8                      b. 3                      c. 4                      d. 6

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Pentru subprogramul **f** definit mai jos, ce se afișează ca urmare a apelului **f(3, 17)**? (6p.)

<pre>//C void f ( int a, int b) { if(a&lt;=b)   {f(a+1,b-2); printf("%c",'*');}   else printf("%d",b); }</pre>	<pre>//C++ void f ( int a, int b) { if(a&lt;=b)   { f(a+1,b-2); cout&lt;&lt;'*';}   else cout&lt;&lt;b; }</pre>
--	---

3. Scrieți un program C/C++ care citește de la tastatură un număr natural **n** cu cel mult 8 cifre ( $n \geq 10$ ) și care creează fișierul text **NR.TXT** ce conține numărul **n** și toate prefixele nenule ale acestuia, pe o singură linie, separate prin câte un spațiu, în ordine descrescătoare a valorii lor.

**Exemplu:** pentru **n=10305** fișierul **NR.TXT** va conține numerele:

10305 1030 103 10 1 (10p.)

4. Subprogramul **f**, cu un parametru:

- primește prin intermediul parametrului **a** un număr natural cu cel mult 8 cifre ( $a > 1$ )  
- returnează cel mai mic divizor prim al valorii parametrului **a**.

**Exemplu:** pentru valoarea **45** a parametrului **a**, subprogramul va returna valoarea **3** deoarece  $a = 3^2 \cdot 5$ , iar cel mai mic divizor prim al său este 3.

**a)** Scrieți definiția completă a subprogramului **f**. (4p.)

**b)** Scrieți un program C/C++ care să citească de la tastatură un număr natural nenul **n** ( $n \leq 100$ ) și apoi un șir de **n** numere naturale de cel mult 8 cifre fiecare, toate numerele din șir fiind strict mai mari decât 1. Folosind apeluri utile ale subprogramului **f**, programul va determina și va afișa pe ecran toate numerele prime din șirul citit. Numerele determinate se vor afișa pe ecran, separate prin câte un spațiu, în ordine crescătoare a valorii lor. Dacă nu există astfel de numere se va afișa pe ecran mesajul **NU EXISTA**.

**Exemplu:** pentru **n=7**, șirul: 1125, 2, 314, 101, 37, 225, 12 pe ecran se va afișa:

2 37 101 (6p.)

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Utilizând metoda backtracking sunt generate numerele de 3 cifre, având toate cifrele distincte și cu proprietatea că cifrele aflate pe poziții consecutive sunt de paritate diferită. Știind că primele șase soluții generate sunt, în această ordine, 103, 105, 107, 109, 123, 125, care este a zecea soluție generată? **(4p.)**
- a. 145                      b. 147                      c. 230                      d. 149

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Se consideră subprogramul alăturat:  
Ce valoare are  $f(15, 2)$ ? Dar  $f(128, 2)$ ? **(6p.)**
- ```
int f(int a, int b){  
    if (b<1) return -1;  
    else  
        if (a%b==0)  
            return 1+f(a/b,b);  
        else  
            return 0; }
```
3. Scrieți un program C/C++ care citește de la tastatură un număr natural  $n$  ( $n \leq 100$ ) și apoi cele  $n$  elemente, numere naturale cu cel mult 4 cifre fiecare, ale unui tablou unidimensional **a**. Programul afișează pe o linie a ecranului suma celor  $n$  elemente ale tabloului, pe următoarea linie a ecranului suma primelor  $n-1$  elemente și așa mai departe, astfel încât ultima linie afișată să conțină doar primul element al tabloului.
- Exemplu:** dacă  $n=4$  iar tabloul are elementele  
 $a=(1, 7, 3, 4)$  programul va afișa valorile alăturate: **(10p.)**
- ```
15  
11  
8  
1
```
4. Se consideră fișierul **BAC.TXT** ce conține un șir **crescător** cu cel mult un milion de numere naturale de cel mult nouă cifre fiecare, separate prin câte un spațiu.
- a)** Să se scrie un program C/C++ care, folosind un algoritm eficient din punct de vedere al memoriei utilizate și al timpului de executare, citește din fișier toți termenii șirului și afișează pe ecran, pe o singură linie, fiecare termen distinct al șirului urmat de numărul de apariții ale acestuia în șir. Valorile afișate sunt separate prin câte un spațiu.
- Exemplu:** dacă fișierul **BAC.TXT** are următorul conținut:
- ```
1 1 1 5 5 5 5 9 9 11 20 20 20
```
- programul va afișa:
- ```
1 3 5 4 9 2 11 1 20 3
```
- deoarece 1 apare de 3 ori, 5 apare de 4 ori, etc. **(6p.)**
- b)** Descrieți succint, în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența ei (3 – 4 rânduri). **(4p.)**

**Subiectul III (30 de puncte)**

Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

1. Se consideră subprogramul `f` definit alăturat. Ce se va afișa în urma apelului `f(12345);`? (4p.)

```
void f(long int n)
{
    if (n!=0)
    {
        if (n%2 == 0)
            cout<<n%10; | printf("%d",n%10);
        f(n/10);
    }
}
```

a. 513

b. 24

c. 42

d. 315

Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.

2. Folosind tehnica backtracking un elev a scris un program care generează toate numerele de câte  $n$  cifre ( $0 < n \leq 9$ ), cifrele fiind în ordine strict crescătoare. Dacă  $n$  este egal cu 5, scrieți în ordine crescătoare toate numerele având cifra unităților 6, care vor fi generate de program. (6p.)
3. Scrieți un program C/C++ care citește de la tastatură un număr natural  $n$  ( $0 < n \leq 100$ ) și cele  $3 \cdot n$  elemente ale tabloului unidimensional  $v$ , fiecare element fiind un număr natural cu cel mult patru cifre fiecare. Tabloul este împărțit în trei zone, cu câte  $n$  elemente: prima zonă conține primele  $n$  elemente din tablou, a doua zonă conține următoarele  $n$  elemente din tablou, restul elementelor fiind în zona a treia. Programul va interschimba primul element par (dacă există) al zonei **unu** cu ultimul element impar (dacă există) al zonei **trei** și apoi va scrie pe prima linie a fișierului text **BAC.TXT** toate elementele tabloului, separate prin câte un spațiu. În cazul în care unul dintre aceste două elemente, care urmează a fi interschimbate, nu există, programul nu va efectua nici o modificare asupra tabloului dat.  
**Exemplu:** pentru  $n=3$  și  $v=(1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9)$ , fișierul **BAC.TXT** va conține:  
**1 9 3 4 5 6 7 8 2** (10p.)

4. Se consideră șirul definit de relația de recurență alăturată:

$$f_n = \begin{cases} n, & \text{dacă } n \leq 5 \\ 2 \cdot f_{n-1}, & \text{dacă } n > 5 \end{cases}$$

a) Scrieți definiția completă a unui subprogram **sub**, care primește prin intermediul singurului său parametru  $n$  un număr natural de maximum 8 cifre, și care returnează cel mai mare termen al șirului  $f$  care este mai mic sau cel mult egal cu  $n$ .

**Exemplu:** dacă  $n=83$  atunci subprogramul va returna valoarea 80. (4p.)

b) Scrieți un program C/C++ care citește de la tastatură un număr natural  $s$  ( $s \leq 10000000$ ) și determină scrierea lui  $s$  ca sumă de termeni distincți ai șirului dat folosind apeluri utile ale subprogramului **sub**. Numerele astfel determinate se vor scrie pe ecran, pe aceeași linie, separate prin câte un spațiu.

**Exemplu:** dacă valoarea citită de la tastatură este 63, se va afișa:

40 20 3

(6p.)

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Utilizând metoda backtracking sunt generate numerele de 3 cifre care au cifrele în ordine crescătoare, iar cifrele aflate pe poziții consecutive sunt de paritate diferită. Știind că primele cinci soluții generate sunt, în această ordine, 123, 125, 127, 129, 145, care este cel de al 8-lea număr generat? (4p.)
- a. 169                      b. 149                      c. 167                      d. 147

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Se consideră subprogramul `f`, descris alăturat. Ce se va afișa în urma apelului `f(3);`? **(6p.)**

```
void f(int n)
{ if (n!=0)
  { if (n%2==0)
    cout<<n<<' '; | printf("%d ",n);
    f(n-1);
    cout<<n<<' '; | printf("%d ",n);
  }
  else cout<<endl; | printf("\n");
}
```

3. Pe prima linie a fișierului text **BAC.TXT** se află o valoare naturală  $n$  ( $1 < n \leq 50$ ), iar pe a doua linie  $n$  numere naturale cu maximum 4 cifre fiecare, despărțite prin câte un spațiu. În șirul numerelor de pe a doua linie a fișierului există cel puțin două numere pătrate perfecte. Scrieți un program C/C++ care citește toate numerele din fișier și afișează pe ecran expresia aritmetică reprezentând suma numerelor de pe a doua linie a fișierului care au proprietatea că sunt pătrate perfecte, cu simbolul  $+$  între ele și, după un semn  $=$ , valoarea acestei sume, ca în exemplu. Termenii sumei afișate se pot afla în orice ordine.

**Exemplu:** dacă fișierul **BAC.TXT** are următorul conținut

5

9 5 36 9 8

atunci pe ecran se poate afișa:

9+9+36=54 sau 9+36+9=54 sau 36+9+9=54

**(10p.)**

4. Subprogramul `sub` primește prin intermediul parametrilor:
- $n$  și  $m$  două numere naturale ( $1 < n < 100$ ,  $1 < m < 100$ )
  - $a$  și  $b$  două tablouri unidimensionale, fiecare având componente numere naturale de maximum patru cifre, **ordonate crescător**; tabloul  $a$  conține  $n$  numere, toate pare, iar tabloul  $b$  conține  $m$  numere, toate impare.

Subprogramul va afișa pe ecran, în ordine crescătoare, separate prin câte un spațiu, un șir format dintr-un număr maxim de elemente care aparțin cel puțin unuia dintre tablouri, astfel încât orice două elemente aflate pe poziții consecutive să fie de paritate diferită.

**Exemplu:** pentru  $n=5$ ,  $m=3$  și tablourile  $a=(2,4,8,10,14)$  și  $b=(3,5,11)$ , subprogramul va afișa 2 3 4 5 8 11 14 sau 2 3 4 5 10 11 14.

**a)** Scrieți definiția completă a subprogramului `sub`, alegând pentru rezolvare un algoritm eficient din punctul de vedere al timpului de executare. **(6p.)**

**b)** Descrieți succint, în limbaj natural, algoritmul pe baza căruia a fost scris subprogramul de la punctul **a)**, explicând în ce constă eficiența metodei utilizate. **(4p.)**



**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

- |   |   |
|---|---|
| <p>1. Se consideră subprogramul <b>f</b> cu definiția alăturată. Ce valoare are <b>f(1213111,1)</b>?<br/><span style="float: right;">(4p.)</span></p> | <pre>int f (long n, int k){<br/>    if (n!=0)<br/>        if (n%10==k)<br/>            return 1+f(n/10,k);<br/>        else return 0;<br/>    else return 0;}</pre> |
| <p>a. 5                                      b. 3</p>   | <p>c. 2                                      d. 1</p>   |

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Utilizând metoda backtracking, sunt generate n ordine crescătoare toate numerele de 3 cifre, astfel încât cifrele sunt în ordine crescătoare, iar cifrele aflate pe poziții consecutive sunt de paritate diferită. Știind că primele trei soluții generate sunt, în această ordine, 123, 125, 127, scrieți toate numerele generate care au suma cifrelor egală cu 12. (6p.)
3. Scrieți definiția completă a subprogramului **sub** cu doi parametri: **n** (număr natural,  $0 < n \leq 50$ ) și **k** (număr natural,  $0 < k \leq 20$ ). Subprogramul determină afișarea pe o linie nouă a ecranului, în ordine descrescătoare, a primelor **n** numere naturale nenule divizibile cu **k**. Numerele vor fi separate prin câte spațiu.  
**Exemplu:** dacă **n=3** și **k=5** la apelul **subprogramului** se va afișa pe ecran:  
15 10 5 (10p.)
4. Se consideră fișierul **BAC.TXT** ce conține cel mult un milion de numere naturale separate prin spații, fiecare număr având cel mult nouă cifre.
- a) Scrieți un program C/C++ care citește toate numerele din fișierul **BAC.TXT** și determină, folosind un algoritm eficient din punct de vedere timpului de executare, cele mai mari două numere de trei cifre care nu se află în fișier. Cele două numere vor fi afișate pe ecran în ordine descrescătoare, cu un spațiu între ele. Dacă nu pot fi determinate două astfel de numere, programul va afișa pe ecran valoarea 0.  
**Exemplu:** dacă fișierul **BAC.TXT** conține numerele:  
12 2345 123 67 989 6 999 123 67 989 999  
atunci programul va afișa  
998 997 (6p.)
- b) Descrieți succint, în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența ei (3 – 4 rânduri). (4p.)

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Se consideră subprogramul cu definiția alăturată. Ce valoare are  $f(3,1)$ ?

(4p.)

```
int f(int n,int y)
{ if(n!=0)
  { y=y+1;
    return y+f(n-1,y);
  }
  else return 0;
}
```

a. 9

b. 6

c. 7

d. 8

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Un elev a scris un program care, folosind metoda backtracking, generează toate numerele de câte 5 cifre, cifrele fiind în ordine strict crescătoare. Scrieți toate numerele generate de program care au prima cifră 5. (6p.)

3. Scrieți definiția completă a subprogramului **sub** cu trei parametri: **n** (număr natural,  $5 < n \leq 30000$ ), **a** și **b**; subprogramul furnizează prin intermediul parametrilor **a** și **b** cele mai mari două numere **prime distincte** mai mici decât **n**.

**Exemplu:** dacă **n= 28** la apelul **subprogramului** se va furniza prin parametrul **a** valoarea 23 și prin parametrul **b** valoarea 19. (10p.)

4. Evidența produselor vândute de o societate comercială este păstrată în fișierul **PRODUSE.TXT**. Pentru fiecare vânzare se cunosc: tipul produsului (un număr natural de cel mult 4 cifre), cantitatea vândută exprimată în kilograme (un număr natural mai mic sau egal cu 100) și prețul unui kilogram (un număr natural mai mic sau egal cu 100).

Fișierul **PRODUSE.TXT** are cel mult 200000 de linii și fiecare linie conține trei numere naturale, separate prin câte un spațiu, ce reprezintă, în această ordine tipul, cantitatea și prețul de vânzare al unui produs la momentul vânzării respective.

**a)** Să se scrie un program **C/C++** care, utilizând un algoritm eficient din punct de vedere al timpului de executare, determină pentru fiecare tip de produs vândut suma totală obținută în urma vânzărilor. Programul va afișa pe câte o linie a ecranului tipul produsului și suma totală obținută, separate prin câte un spațiu, ca în exemplu.

**Exemplu:** dacă fișierul **PRODUSE.TXT** are conținutul alăturat, programul va afișa perechile următoare, nu neapărat în această ordine:

1 150

2 30

3 5

```
3 1 5
1 20 5
2 10 3
1 10 5
```

(6p.)

**b)** Descrieți succint, în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența (3 - 4 rânduri). (4p.)

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Un algoritm de tip backtracking generează, în ordine lexicografică, toate șirurile de 5 cifre 0 și 1 cu proprietatea că nu există mai mult de două cifre 0 pe poziții consecutive. Primele 7 soluții generate sunt: 00100, 00101, 00110, 00111, 01001, 01010, 01011. Care este a 8-a soluție generată de acest algoritm? **(4p.)**
- a. 01110                      b. 01100                      c. 01011                      d. 01101

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Pentru funcția  $f$  definită alăturat, stabiliți care este valoarea  $f(5)$ . Dar  $f(23159)$ ? **(6p.)**
- ```
int f(int n){
    int c;
    if (n==0) return 9;
    else
    {c=f(n/10);
      if (n%10<c) return n%10;
      else return c;
    }
}
```
3. Fișierul text **numere.txt** conține pe prima linie un număr natural  $n$  ( $n < 30000$ ), iar pe a doua linie  $n$  numere întregi având maximum 4 cifre fiecare. Se cere să se afișeze pe ecran un șir de  $n$  numere întregi, cu proprietatea că valoarea termenului de pe poziția  $i$  ( $i=1, 2, \dots, n$ ) din acest șir este egală cu cea mai mare dintre primele  $i$  valori de pe a doua linie a fișierului **numere.txt**.
- a) Descrieți pe scurt un algoritm de rezolvare, eficient din punct de vedere al timpului de executare și al spațiului de memorie utilizat, explicând în ce constă eficiența sa. **(4p.)**
- b) Scrieți programul C/C++ corespunzător algoritmului descris. **(6p.)**
- Exemplu:** dacă fișierul **numere.txt** are conținutul
- |                           |                          |
|---------------------------|--------------------------|
| 12                        | 12                       |
| 4 6 3 7 8 1 6 2 7 9 10 8  | 4 6 3 7 8 1 6 2 7 9 10 8 |
| 4 6 6 7 8 8 8 8 8 9 10 10 |                          |
4. a) Scrieți doar antetul funcției **sum** care primește ca parametru un număr natural nenul  $x$  cu maximum 9 cifre și returnează suma divizorilor numărului  $x$ . **(3p.)**
- Exemplu:** **sum(6)** are valoarea 12 ( $=1+2+3+6$ ).
- b) Să se scrie un program C/C++ care citește de la tastatură un număr natural  $n$  ( $0 < n < 25$ ) și apoi  $n$  numere naturale nenule cu maximum 9 cifre fiecare. Programul calculează, folosind apeluri ale funcției **sum**, și afișează pe ecran câte numere prime conține șirul citit.
- Exemplu:** pentru  $n=5$  și valorile 12 3 9 7 1 se va afișa pe ecran valoarea 2 (în șirul dat există două numere prime și anume 3 și 7). **(7p.)**

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

- |                                                                                                                                                                                                                   |                                              |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|
| <p>1. Se consideră subprogramul <b>P</b>, definit alăturat. Știind că valoarea variabilei întregi <b>a</b> este înainte de apel <b>4</b>, care este valoarea ei imediat după apelul <b>P(a)</b>? <b>(4p.)</b></p> | <pre>void P(int &amp;x) {     x=x+5; }</pre> |
| <p>a. 10                      b. 4                      c. 9                      d. 5</p>                                                                                                                        |                                              |

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Pentru a scrie valoarea **10** ca sumă de numere prime se folosește metoda backtracking și se generează, în această ordine, sumele distincte: **2+2+2+2+2**, **2+2+3+3**, **2+3+5**, **3+7**, **5+5**. Folosind exact aceeași metodă, se scrie valoarea **9** ca sumă de numere prime. Care sunt primele trei soluții, în ordinea generării lor? **(6p.)**
3. Fișierele text **NR1.TXT** și **NR2.TXT** conțin, separate prin câte un spațiu, mai multe numere întregi de cel mult **9** cifre fiecare. Fiecare dintre fișiere conține cel mult **100** de valori și numerele din fiecare fișier sunt ordonate strict crescător. Se cere să se afișeze pe ecran, în ordine crescătoare, numerele divizibile cu **5** care se găsesc doar în unul din cele două fișiere.
- Exemplu:** dacă fișierul **NR1.TXT** conține numerele **1 2 3 4 7 20 60**, iar fișierul **NR2.TXT** conține numerele **3 5 7 8 9 10 12 20 24**, atunci se vor afișa pe ecran valorile **5 10 60**.
- a) Descrieți un algoritm de rezolvare a acestei probleme, eficient din punct de vedere al timpului de executare și al spațiului de memorie utilizat, explicând în ce constă eficiența acestuia. **(4p.)**
- b) Scrieți programul **C/C++** corespunzător algoritmului descris. **(6p.)**
4. Scrieți un program **C/C++** care citește de la tastatură o valoare naturală nenulă **n** (**n ≤ 20**), apoi un șir de **n** numere naturale, având fiecare exact **5** cifre. Dintre cele **n** numere citite, programul determină pe acelea care au toate cifrele egale și le afișează pe ecran, în ordine crescătoare, separate prin câte un spațiu.
- Exemplu:** pentru **n=5** și numerele **11111 33333 12423 59824 11111 33443** se va afișa: **11111 11111 33333**. **(10p.)**

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Fie subprogramul `fct` definit alăturat, parțial. Inițial, variabile întregi `a`, `b` și `c` au valorile `a=8`, `b=31` și `c=9`, iar după apelul `fct(a,b,c)`, valorile celor trei variabile sunt `a=9`, `b=31` și `c=39`. Care poate fi antetul subprogramului `fct`? (4p.)
- |                                                          |                                                          |
|----------------------------------------------------------|----------------------------------------------------------|
| <pre>void fct(int &amp;x,int &amp;y,int &amp;z)</pre>    | <pre>void fct(....) {   x=x+1;   y=y-1;   z=x+y; }</pre> |
| <pre>a. void fct(int &amp;x,int &amp;y,int &amp;z)</pre> | <pre>b. void fct(int x,int &amp;y,int &amp;z)</pre>      |
| <pre>c. void fct(int x,int y,int z)</pre>                | <pre>d. void fct(int &amp;x,int y,int &amp;z)</pre>      |

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Trei băieți, **Alin**, **Bogdan** și **Ciprian**, și trei fete, **Delia**, **Elena** și **Felicia**, trebuie să formeze o echipă de 3 copii, care să participe la un concurs. Echipa trebuie să fie mixtă (adică să conțină cel puțin o fată și cel puțin un băiat). Ordinea copiilor în echipă este importantă deoarece aceasta va fi ordinea de intrare a copiilor în concurs (de exemplu echipa **Alin, Bogdan, Delia** este diferită de echipa **Bogdan, Alin, Delia**).
- Câte echipe se pot forma, astfel încât din ele să facă parte simultan **Alin** și **Bogdan**?
  - Dați exemplu de o echipă corect formată din care să nu facă parte nici **Alin** și nici **Bogdan**. (6p.)
3. Se consideră șirul `1, 2,1, 3,2,1, 4,3,2,1, ...` construit astfel: prima grupă este formată din numărul 1, a doua grupă este formată din numerele 2 și 1, iar grupa a  $k$ -a, este formată din numerele  $k, k-1, \dots, 1$ . Se cere să se citească de la tastatură un număr natural  $n$  ( $n \leq 1000$ ) și să se afișeze pe ecran cel de al  $n$ -lea termen al șirului dat.
- a) Descrieți un algoritm de rezolvare a acestei probleme, eficient din punct de vedere al timpului de executare și al spațiului de memorie, explicând în ce constă eficiența acestuia. (4p.)
- b) Scrieți programul C/C++ corespunzător algoritmului descris (6p.)
4. Se consideră subprogramul `P` care are doi parametri:
- `n`, prin intermediul căruia primește un număr natural de cel mult 9 cifre
  - `c`, prin intermediul căruia primește o cifră.
- Subprogramul va furniza tot prin intermediul parametrului `n` numărul obținut din `n` prin eliminarea tuturor aparițiilor cifrei `c`. Dacă, după eliminare, numărul nu mai conține nicio cifră sau conține doar cifre 0, rezultatul returnat va fi 0.
- a) Scrieți doar antetul subprogramului `P`. (2p.)
- b) Pe prima linie a fișierului text **BAC.IN** se găsesc, separate prin câte un spațiu, mai multe numere naturale de cel mult 9 cifre fiecare. Scrieți programul C/C++ care citește numerele din acest fișier, utilizând apeluri ale subprogramului `P` elimină toate cifrele impare din fiecare dintre aceste numere și apoi scrie în fișierul text **BAC.OUT** numerele astfel obținute, separate prin câte un spațiu. Dacă un număr din fișierul **BAC.IN** nu conține nicio cifră pară nenulă, acesta nu va mai apărea deloc în fișierul de ieșire. (8p.)
- Exemplu:** dacă fișierul **BAC.IN** conține numerele 25 7 38 1030 45127 0 35 60 15, atunci **BAC.OUT** va avea conținutul: 2 8 42 60.

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Utilizând metoda backtracking se generează permutările cuvântului **info**. Dacă primele trei soluții generate sunt: **fino**, **fion**, **fnio** care este cea de-a cincea soluție? **(4p.)**
- a. **foin**                      b. **fnoi**                      c. **foni**                      d. **ifon**

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Pentru funcțiile **f1** și **f2** definite alăturat, stabiliți care este valoarea lui **f1(3)**. Dar **f2(41382)**? **(6p.)**
- ```
long f1(int c)
{ if (c%2==1) return 1;
  else return 2;
}

long f2(long n)
{ if (n==0) return 0;
  else return f1(n%10)+f2(n/10);
}
```
3. Se citește de la tastatură un număr natural **n** ( $n \leq 500$ ) și apoi **n** cifre separate prin spații. Se cere să se afișeze pe ecran cele **n** cifre citite, în ordine crescătoare, separate prin câte un spațiu.  
**Exemplu:** pentru **n=19** și cifrele 3 3 0 9 2 1 2 1 3 7 1 5 2 7 1 0 3 2 3 se va afișa pe ecran 0 0 1 1 1 1 2 2 2 2 3 3 3 3 3 5 7 7 9.
- a) Descrieți pe scurt un algoritm de rezolvare al problemei, eficient din punct de vedere al spațiului de memorie utilizat și al timpului de executare, explicând în ce constă eficiența metodei alese. **(4p.)**
- b) Scrieți programul C/C++ corespunzător algoritmului descris. **(6p.)**
4. Fișierul text **BAC.TXT** conține mai multe numere naturale, cu cel mult 6 cifre fiecare, câte un număr pe fiecare linie a fișierului.  
Scrieți un program C/C++ care citește toate numerele din fișierul **BAC.TXT** și le afișează pe ecran, în aceeași ordine, câte **cinci** pe fiecare linie, separate prin câte un spațiu, cu excepția ultimei linii care poate conține mai puțin de cinci numere. Programul va afișa apoi pe ecran, pe o linie separată, câte numere din fișier au suma cifrelor pară.
- Exemplu:** dacă fișierul are conținutul alăturat, pe ecran se vor afișa numerele de mai jos:
- |    |    |
|----|----|
| 11 | 11 |
| 21 | 21 |
| 30 | 30 |
| 40 | 40 |
| 51 | 51 |
| 16 | 16 |
| 17 | 17 |
| 10 | 10 |
| 1  | 1  |
| 4  |    |
- (10p.)**

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Câte numere cu exact două cifre pot fi construite folosind doar cifre pare distincte? **(4p.)**  
a. 12                                      b. 16                                      c. 20                                      d. 25

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Pentru funcțiile **f** și **g** definite mai jos, scrieți care este rezultatul returnat la apelul **g(11)**. Dar rezultatul returnat la apelul **f(6)**? **(6p.)**

<pre>long g(long x) { if (x&gt;9)     return (x/10 + x%10);   else     return x; }</pre>	<pre>long f(int c) { if (c&lt;1)     return 1;   else     return g(c+f(c-1)); }</pre>
--	---

3. Scrieți un program C/C++ care citește de la tastatură un număr natural **n** ( $n \leq 32000$ ) și afișează pe ecran numărul natural din intervalul închis **[1,n]** care are cei mai mulți divizori. Dacă există mai multe numere cu această proprietate se va afișa cel mai mic dintre ele.

**Exemplu:** pentru **n=20** se va afișa valoarea **12** (12, 18 și 20 au câte 6 divizori, iar 12 este cel mai mic dintre ele). **(10p.)**

4. În fișierul text **BAC.IN** se găsesc, pe o singură linie, separate prin câte un spațiu, mai multe numere naturale de cel mult 6 cifre fiecare. Se cere să se determine și să se afișeze pe ecran, separate printr-un spațiu, ultimele **două** numere impare (nu neapărat distincte) din fișierul **BAC.IN**. Dacă în fișier se găsește un singur număr impar sau niciun număr impar se va scrie pe ecran mesajul **Numere insuficiente**.

**Exemplu:** dacă fișierul **BAC.IN** conține valorile: 12 15 68 13 17 90 31 42 se va afișa 17 31.

**a)** Descrieți în limbaj natural un algoritm eficient din punct de vedere al spațiului de memorie și al timpului de executare, pentru rezolvarea acestei probleme, explicând în ce constă eficiența acestuia. **(4p.)**

**b)** Scrieți programul C/C++ corespunzător algoritmului descris. **(6p.)**

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Funcția  $F$  are definiția alăturată. Ce valoare are  $F(3)$ ? (4p.)

```
int F(int n)
{if(n==0 || n==1) return 1;
 else
  return 2*F(n-1)+2*F(n-2);}
```

a. 1

b. 12

c. 6

d. 10

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare**

2. Un algoritm generează în ordine crescătoare toate numerele de  $n$  cifre, folosind doar cifrele 3, 5 și 7. Dacă pentru  $n=5$ , primele cinci soluții generate sunt 33333, 33335, 33337, 33353, 33355, precizați care sunt ultimele **trei** soluții generate, în ordinea generării. (6p.)
3. Scrieți definiția completă a subprogramului **multiplu** care are 3 parametri:  $a$ , prin care primește un tablou unidimensional cu maximum 100 de numere naturale mai mici decât 1000,  $n$ , numărul efectiv de elemente ale tabloului și  $k$ , un număr natural ( $k \leq 9$ ). Subprogramul returnează numărul de elemente din tablou care sunt multipli ai numărului  $k$  și au ultima cifră egală cu  $k$ .  
**Exemplu:** dacă  $n=6$ ,  $a=(9, \underline{273}, \underline{63}, 83, \underline{93}, \underline{123})$ , iar  $k=3$ , subprogramul va returna valoarea 4. (10p.)
4. În fișierul **numere.txt** sunt memorate maximum 10000 de numere naturale cu cel mult 9 cifre fiecare. Fiecare linie a fișierului conține câte un număr. Se cere afișarea pe ecran, în ordine descrescătoare, a tuturor cifrelor care apar în numerele din fișier. Alegeți un algoritm de rezolvare eficient din punct de vedere al timpului de executare.  
**Exemplu:** dacă fișierul **numere.txt** conține:  
267  
39628  
79  
se va tipări 9987766322.
- a) Descrieți succint, în limbaj natural, strategia de rezolvare și justificați eficiența algoritmului ales. (4p.)
- b) Scrieți programul C/C++ corespunzător algoritmului ales. (6p.)



**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

- |  |   |
|--|---|
| <p>1. Subprogramul <code>f</code> are definiția alăturată. Ce se va afișa în urma apelului <code>f(12345);</code>?<br/><br/><b>(4p.)</b></p> | <pre>void f(long n) {     if (n&gt;9)     {         cout&lt;&lt;n/100;   printf("%d",n/100);         f(n/10);     } }</pre> |
| <p>a. 1231210                      b. 123121</p>   | <p>c. 1234123121                      d. 123</p>  |

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare**

2. Un algoritm generează în ordine descrescătoare toate numerele de 5 cifre, fiecare dintre ele având cifrele în ordine strict crescătoare. Știind că primele cinci soluții generate sunt **56789**, **46789**, **45789**, **45689**, **45679**, precizați care sunt ultimele **trei** soluții generate, în ordinea generării. **(6p.)**
3. Scrieți definiția completă a subprogramului **interval** care are doi parametri **a** și **n**, prin care primește un tablou unidimensional cu maximum 100 de numere naturale mai mici decât 1000 și respectiv numărul efectiv de elemente din tabloul unidimensional. Subprogramul returnează numărul de elemente din tabloul unidimensional care aparțin intervalului închis determinat de primul și respectiv ultimul element al tabloului.  
**Exemplu:** dacă tabloul are 6 elemente și este de forma **(12,27,6,8,9,2)**, subprogramul va returna valoarea 5. **(10p.)**
4. În fișierul **numere.txt** pe prima linie este memorat un număr natural **n** ( $n \leq 10000$ ), iar pe linia următoare un șir de **n** numere naturale distincte două câte două, separate prin câte un spațiu, cu maximum 4 cifre fiecare. Se cere afișarea pe ecran a poziției pe care s-ar găsi primul element din șirul aflat pe linia a doua a fișierului, în cazul în care șirul ar fi ordonat crescător. Numerotarea pozițiilor elementelor în cadrul șirului este de la 1 la **n**. Alegeți un algoritm de rezolvare eficient din punct de vedere al memoriei utilizate și al timpului de executare.  
**Exemplu:** dacă fișierul **numere.txt** conține:  
6  
267 13 45 628 7 79  
se va afișa 5, deoarece primul element din șirul inițial, 267, s-ar găsi pe poziția a cincea în șirul ordonat crescător (7 13 45 79 267 628).  
**a)** Descrieți succint, în limbaj natural, strategia de rezolvare și justificați eficiența algoritmului ales. **(4p.)**  
**b)** Scrieți programul C/C++ corespunzător algoritmului ales. **(6p.)**

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Funcția `f` are definiția alăturată. Ce se va afișa în urma apelului `f(12345,0);`? **(4p.)**

```
void f(long n, int i)
{
    if (i < n % 10)
    {
        cout << n % 10; | printf("%d", n % 10);
        f(n / 10, i + 1);
    }
}
```

a. 54321

b. 543

c. 54

d. 5432

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare**

2. Un algoritm generează, în ordine lexicografică, toate șirurile alcătuite din câte  $n$  cifre binare (0 și 1). Știind că pentru  $n=5$ , primele patru soluții generate sunt 00000, 00001, 00010, 00011, precizați care sunt ultimele **trei** soluții generate, în ordinea obținerii lor. **(6p.)**
3. Scrieți definiția completă a subprogramului `count` care are doi parametri,  $a$  și  $n$ , prin care primește un tablou unidimensional cu maximum 100 de numere reale și respectiv numărul efectiv de elemente din tablou. Subprogramul returnează numărul de elemente din tabloul  $a$  care sunt mai mari sau cel puțin egale cu media aritmetică a tuturor elementelor din tablou.  
**Exemplu:** dacă tabloul are 6 elemente și este de forma (12, 7.5, 6.5, 3, 8.5, 7.5), subprogramul va returna valoarea 4 (deoarece media tuturor elementelor este 7.5 și numerele subliniate sunt cel puțin egale cu această medie). **(10p.)**
4. În fișierul `numere.txt` este memorat un șir de maximum 10000 numere naturale, distincte două câte două, cu maximum 4 cifre fiecare, separate prin câte un spațiu. Pentru un număr  $k$  citit de la tastatură, se cere afișarea pe ecran a poziției pe care se va găsi acesta în șirul de numere din fișier, dacă șirul ar fi ordonat descrescător, sau mesajul **nu există**, dacă numărul  $k$  nu se află printre numerele din fișier. Alegeți un algoritm eficient de rezolvare din punct de vedere al memoriei utilizate și al timpului de executare.  
**Exemplu:** dacă fișierul `numere.txt` conține numerele 26 2 5 30 13 45 62 7 79, iar  $k$  are valoarea 13, se va afișa 6 deoarece 13 s-ar găsi pe poziția a șasea în șirul ordonat descrescător (79 62 45 30 26 13 7 5 2).  
**a)** Descrieți succint, în limbaj natural, strategia de rezolvare și justificați eficiența algoritmului ales. **(4p.)**  
**b)** Scrieți programul C/C++ corespunzător algoritmului ales. **(6p.)**

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Funcția  $F$  are definiția alăturată. Ce valoare are  $F(18)$ ? (4p.)

```
int F(int x){  
    if (x<=1) return x;  
    else return x+F(x-2);  
}
```

a. 90

b. 171

c. 91

d. 18

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare**

2. Un algoritm generează în ordine crescătoare, toate numerele de  $n$  cifre ( $n < 9$ ), cu cifre distincte, care nu au două cifre pare alăturate. Dacă pentru  $n=5$ , primele cinci soluții generate sunt 10325, 10327, 10329, 10345, 10347, precizați care sunt următoarele **trei** soluții generate, în ordinea obținerii lor. (6p.)

3. Subprogramul **aranjare** are doi parametri:  $a$  prin care primește un tablou unidimensional cu maximum 100 de numere reale nenule și  $n$ , numărul de elemente din tablou. Subprogramul rearanjează elementele tabloului unidimensional astfel încât toate valorile negative să se afle pe primele poziții, iar valorile pozitive în continuarea celor negative. Ordinea în cadrul secvenței de elemente pozitive, respectiv în cadrul secvenței de elemente negative, poate fi oricare. Tabloul modificat va fi furnizat tot prin intermediul parametrului  $a$ .

**Exemplu:** dacă tabloul are 6 elemente și este de forma (12, -7.5, 6.5, -3, -8, 7.5), după apel, acesta ar putea fi: (-7.5, -3, -8, 12, 6.5, 7.5).

Scrieți definiția completă a subprogramului **aranjare**.

(10p.)

4. În fișierul **nr1.txt** este memorată pe prima linie o valoare naturală  $n$  de cel mult 8 cifre, iar pe linia următoare sunt memorate  $n$  numere naturale, cu maximum 4 cifre fiecare, ordonate strict crescător și separate prin câte un spațiu. În fișierul **nr2.txt** este memorată pe prima linie o valoare naturală  $m$  de cel mult 8 cifre, iar pe linia următoare sunt memorate  $m$  numere naturale, cu maximum 4 cifre fiecare, ordonate strict crescător și separate prin câte un spațiu. Se cere afișarea pe ecran, separate prin câte un spațiu, în ordine strict crescătoare, a tuturor numerelor aflate pe a doua linie în cel puțin unul dintre cele două fișiere. În cazul în care un număr apare în ambele fișiere, el va fi afișat o singură dată. Alegeți un algoritm de rezolvare eficient din punct de vedere al memoriei utilizate și al timpului de executare.

**Exemplu:** pentru următoarele fișiere:

**nr1.txt**

5

3 6 8 9 12

se va afișa 2 3 5 6 7 8 9 12 13.

**nr2.txt**

6

2 3 5 7 9 13

- a) Descrieți succint, în limbaj natural, strategia de rezolvare și justificați eficiența algoritmului ales. (4p.)

- b) Scrieți programul C/C++ corespunzător algoritmului ales. (6p.)

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Funcția **F** are definiția alăturată. Ce valoare are **F(5)**? (4p.)

```
int F(int x)
{if(x!=0) return x+F(x-1);
 else
  return x;
}
```

a. 5

b. 10

c. 15

d. 6

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare**

2. Un algoritm generează în ordine descrescătoare, toate numerele de **n** cifre (**n < 9**), cu cifrele în ordine strict crescătoare, care nu au două cifre pare alăturate. Dacă pentru **n=5**, primele cinci soluții generate sunt **56789**, **45789**, **45679**, **45678**, **36789**, precizați care sunt următoarele **trei** soluții generate, în ordinea obținerii lor. (6p.)

3. Subprogramul **nule** are doi parametri: **a**, prin care primește un tablou unidimensional cu maximum 100 de numere întregi, cu cel mult 4 cifre fiecare și **n**, numărul de elemente din tablou. Subprogramul rearanjează elementele tabloului unidimensional astfel încât toate valorile nule să se afle la sfârșitul tabloului. Ordinea în cadrul secvenței de elemente nenule poate fi oricare. Tabloul modificat este furnizat tot prin parametrul **a**.

**Exemplu:** dacă **n=6**, **a=(12,0,0,-3,-8,0)**, după apel, acesta ar putea fi:

**a=(12,-3,-8,0,0,0)**.

Scrieți definiția completă a subprogramului **nule**.

(10p.)

4. În fișierul **nr1.txt** este memorată pe prima linie o valoare naturală **n** de cel mult 8 cifre, iar pe linia următoare sunt memorate **n** numere naturale, cu maximum 4 cifre fiecare, ordonate strict crescător și separate prin câte un spațiu. În fișierul **nr2.txt** este memorată pe prima linie o valoare naturală **m** de cel mult 8 cifre, iar pe linia următoare sunt memorate **m** numere naturale, cu maximum 4 cifre fiecare, ordonate strict crescător și separate prin câte un spațiu. Se cere afișarea pe ecran, separate prin câte un spațiu, în ordine strict crescătoare, a tuturor numerelor aflate pe a doua linie atât în primul cât și în al doilea fișier. Alegeți un algoritm de rezolvare eficient din punct de vedere al memoriei utilizate și al timpului de executare.

**Exemplu:** pentru următoarele fișiere:

**nr1.txt**

5

3 6 8 9 12

se va afișa 3 9.

**nr2.txt**

6

2 3 5 7 9 13

- a) Descrieți succint, în limbaj natural, strategia de rezolvare și justificați eficiența algoritmului ales. (4p.)

- b) Scrieți programul C/C++ corespunzător algoritmului ales. (6p.)

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Următoarele probleme se referă la mulțimea de numere reale  $M = \{x_1, x_2, \dots, x_n\}$  ( $1000 < n \leq 10000$ ). Care dintre acestea, comparativ cu celelalte, admite un algoritm care se încheie după un număr minim de pași? **(4p.)**
- a. sortarea elementelor mulțimii  $M$                       b. generarea elementelor produsului cartezian  $M \times M$
- c. determinarea elementului minim al mulțimii  $M$                       d. generarea tuturor permutărilor mulțimii  $M$

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Se consideră subprogramul,  $f$ , definit alăturat.
- Ce valoare are  $f(100)$ ?
  - Scrieți o valoare pentru  $x$  astfel încât  $f(x) = 1$  **(6p.)**

```
int f(int n)
{ if(n==0) return 0;
  else return n%2+f(n/2);
}
```
3. Scrieți definiția completă a unui subprogram `i_prim` care primește prin singurul său parametru,  $n$ , un număr natural din intervalul  $[2, 30000]$  și returnează diferența minimă  $p_2 - p_1$  în care  $p_1$  și  $p_2$  sunt numere prime și  $p_1 \leq n \leq p_2$ .  
**Exemplu:** dacă  $n=20$  atunci  $i\_prim(n)=4$ , valoare obținută pentru  $p_1=19$  și  $p_2=23$ . **(10p.)**
4. Fișierul text `BAC.TXT` conține pe prima linie două numere naturale  $n$  și  $k$  separate de un spațiu ( $3 \leq n \leq 10000$ ,  $2 \leq k \leq n/2$ ), iar pe a doua linie un șir de  $n$  numere naturale  $x_1, x_2, \dots, x_n$  separate prin câte un spațiu, fiecare număr din acest șir având cel mult patru cifre.
- a) Scrieți un program C/C++ care citește numerele din fișier și determină, utilizând o metodă eficientă din punct de vedere al timpului de executare, cel mai mic indice  $i$  ( $1 \leq i \leq n-k+1$ ) pentru care media aritmetică a numerelor  $x_i, x_{i+1}, \dots, x_{i+k-1}$  este maximă. Programul afișează valoarea lui  $i$  pe ecran.  
**Exemplu:** pentru fișierul alăturat se afișează 2, deoarece media maximă se obține pentru 9, 4, 7. **(6p.)**
- ```
8 3
2 9 4 7 5 2 9 9
```
- b) Explicați succint, în limbaj natural, metoda utilizată la punctul a, justificând eficiența acesteia. **(4p.)**



**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Algoritmul de generare a tuturor numerelor de 5 cifre nenule, fiecare având cifrele ordonate strict crescător, este echivalent cu algoritmul de generare a: **(6p.)**
- a. submulțimilor unei mulțimi cu 5 elemente      b. produsului cartezian a unor mulțimi de cifre
- c. aranjamentelor de 9 elemente luate câte 5      d. combinațiilor de 9 elemente luate câte 5

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Pentru subprogramul `suma` definit alăturat, scrieți valoarea expresiei `suma(5,4)`. **(4p.)**
- ```
int suma (int a,int b)
{
    if (a==0 && b==0) return 0;
    else if (a==0) return 1+suma(a,b-1);
    else return 1+suma(a-1,b);
}
```
3. a) Scrieți definiția completă a subprogramului `shift` care primește prin intermediul parametrului `n` o valoare naturală nenulă ( $n \leq 100$ ), iar prin intermediul parametrului `x`, un tablou unidimensional cu maximum 100 de componente. Fiecare componentă a acestui tablou este un număr întreg care are cel mult 4 cifre. Subprogramul permută circular cu o poziție spre stânga primele `n` elemente ale tabloului `x` și furnizează tabloul modificat tot prin parametrul `x`.  
**Exemplu:** dacă înainte de apel `n=4` și `x=(1,2,3,4)`, după apel `x=(2,3,4,1)`. **(4p.)**
- b) Scrieți un program C/C++ care citește de la tastatură o valoare naturală nenulă `n` ( $n \leq 100$ ), apoi cele `n` elemente ale unui tablou unidimensional `x`. Programul va inversa ordinea elementelor tabloului `x` folosind apeluri utile ale subprogramului `shift` și va afișa pe ecran, separate prin câte un spațiu, elementele tabloului rezultat în urma acestei prelucrări.  
**Exemplu:** dacă se citește pentru `n` valoarea 5, iar tabloul `x` este `(1,2,3,4,5)` programul va determina ca `x` să devină `(5,4,3,2,1)`. **(6p.)**
4. Fișierul text `BAC.TXT` conține pe prima linie un număr natural nenul `n` ( $1 \leq n \leq 1000$ ), iar pe fiecare dintre următoarele `n` linii, câte două numere întregi `a` și `b` ( $1 \leq a \leq b \leq 32000$ ), fiecare pereche reprezentând un interval închis de forma `[a,b]`. Scrieți un program C/C++ care determină intervalele care au proprietatea că intersecția cu oricare dintre celelalte `n-1` intervale este vidă și afișează pe câte o linie a ecranului, separate printr-un spațiu, numerele care reprezintă capetele intervalelor determinate. Dacă nu există nici un astfel de interval, se afișează pe ecran mesajul `NU EXISTA`. **(10p.)**

**Exemplu:** dacă fișierul `BAC.TXT` are conținutul alăturat, pe ecran se va afișa:

2	6	sau	17	20
17	20		2	6

4
17 20
2 6
10 15
8 16

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Generând șirurile de maximum 3 caractere distincte din mulțimea  $\{A, B, C, D, E\}$ , ordonate lexicografic, obținem succesiv: A, AB, ABC, ABD, ... . Ce șir va fi generat imediat după BAE? (4p.)
- a. BCA b. CAB  
c. BC d. BEA

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Funcția  $f$  are definiția alăturată. (3p.)
- a) Ce valoare are  $f(17)$ ? (3p.)
- b) Ce valoare are  $f(22)$ ? (3p.)
- ```
int f(int n)
{
    if (n <= 9) return 0;
    if (n % 4 == 0) return 0;
    return 1 + f(n - 3);
}
```
3. a) Scrieți definiția completă a subprogramului  $p$  care primește prin intermediul parametrului  $n$  un număr natural nenul ( $n \leq 100$ ), iar prin intermediul parametrului  $x$  un tablou unidimensional cu  $n$  componente întregi, de maximum patru cifre fiecare. Subprogramul furnizează prin intermediul parametrului  $mini$  valoarea minimă din tabloul  $x$ , prin intermediul parametrului  $maxi$  valoarea maximă din  $x$ , iar prin intermediul parametrului  $sum$  suma elementelor din tabloul  $x$ . (6p.)
- b) Scrieți un program C/C++ care citește de la tastatură o valoare naturală nenulă  $n$ , ( $3 \leq n \leq 100$ ), apoi cele  $n$  elemente distincte ale unui tablou unidimensional  $x$ . Fiecare dintre aceste elemente este un număr natural având cel mult patru cifre. Folosind apeluri utile ale subprogramului  $p$ , programul calculează și afișează pe ecran media aritmetică a elementelor care ar rămâne în tabloul  $x$  dacă s-ar elimina valoarea minimă și valoarea maximă din tablou. Valoarea afișată va avea cel mult 3 cifre după virgulă.  
**Exemplu:** dacă se citesc pentru  $n$  valoarea 5, iar pentru tabloul  $x$  valorile (1, 9, 4, 8, 5), programul va afișa una dintre valorile 5.667 sau 5.666. (4p.)
4. Fișierul text `bac.txt` conține pe prima linie numărul natural  $n$ ,  $1 \leq n \leq 30000$ , pe următoarele  $n$  linii un șir de  $n$  numere întregi, ordonate crescător, iar pe ultima linie două numere întregi  $a$  și  $b$  ( $a \leq b$ ) separate de un spațiu. Fiecare dintre cele  $n$  numere, precum și valorile  $a$  și  $b$ , au cel mult patru cifre.
- a) Scrieți un program C/C++, eficient din punct de vedere al timpului de executare, care afișează pe ecran cel mai mic număr întreg din intervalul închis  $[a, b]$  care se găsește în șirul dat. Dacă nu există un astfel de număr, programul afișează textul `NU`.  
**Exemplu:** dacă fișierul `bac.txt` are conținutul alăturat, programul afișează valoarea 11 (6p.)
- b) Descrieți în limbaj natural metoda utilizată și explicați în ce constă eficiența ei. (4p.)
- ```
4
-2
7
11
35
8 15
```



**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Un program citește o valoare naturală nenulă impară pentru  $n$  și apoi generează și afișează în ordine crescătoare lexicografic toate combinațiile formate din  $n$  cifre care îndeplinesc următoarele proprietăți:
- încep și se termină cu 0;
  - modulul diferenței între oricare două cifre alăturate dintr-o combinație este 1.
- Astfel, pentru  $n=5$ , combinațiile afișate sunt, în ordine, următoarele: 01010, 01210. Dacă se rulează acest program și se citește pentru  $n$  valoarea 7, imediat după combinația 0101210 va fi afișată combinația: **(4p.)**
- a. 0121210                      b. 0123210                      c. 0111210                      d. 0121010

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Funcția  $f$  are definiția alăturată:
- a) Ce valoare are  $f(16)$ ? **(3p.)**
- b) Scrieți cea mai mare valoare de două cifre pe care o poate avea  $n$  astfel încât  $f(n)$  să fie egal cu 2. **(3p.)**
- ```
int f(int n)
{
    if (n<=0) return -1;
    if (n%2==0) return 0;
    if (n%3==0) return 0;
    return 1+f(n-10);
}
```
3. Subprogramul  $f$  primește prin intermediul parametrului  $n$  un număr natural nenul ( $1 \leq n \leq 9$ ), iar prin intermediul parametrului  $a$ , un tablou unidimensional care conține  $n$  valori naturale, fiecare dintre acestea reprezentând câte o cifră a unui număr. Astfel,  $a_0$  reprezintă cifra unităților numărului,  $a_1$  cifra zecilor etc. Subprogramul furnizează prin parametrul  $k$  o valoare naturală egală cu numărul obținut din cifrele pare reținute în tabloul  $a$  sau valoarea -1 dacă în tablou nu există nicio cifră pară. Scrieți definiția completă a subprogramului  $f$ .
- Exemple:** dacă subprogramul se apelează pentru  $n=6$  și pentru tabloul  $a$  având valorile (2,3,5,6,4,1), parametrul  $k$  va furniza valoarea 462. Dacă subprogramul se apelează pentru  $n=4$  și pentru  $a$  reținând valorile (0,0,1,1),  $k$  va furniza valoarea 0. Dacă subprogramul se apelează pentru  $n=3$  și pentru  $a$  reținând valorile (3,7,1),  $k$  va furniza valoarea -1. **(10p.)**
4. Fișierul text **NUMAR.TXT** conține pe prima linie un număr real pozitiv  $x$  care are cel mult două cifre la partea întreagă și cel mult șapte cifre după punctul zecimal..
- a) Scrieți un program C/C++ care, utilizând un algoritm eficient din punct de vedere al timpului de executare și al memoriei utilizate, afișează pe ecran, separate printr-un spațiu, două numere naturale al căror raport este egal cu  $x$  și a căror diferență absolută este minimă.
- Exemplu:** dacă fișierul conține valoarea alăturată, se vor afișa pe ecran
- |       |              |
|-------|--------------|
| 0.375 | <b>(6p.)</b> |
|-------|--------------|
- numerele 3 8.
- b) Descrieți în limbaj natural metoda utilizată și explicați în ce constă eficiența ei. **(4p.)**

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Pentru generarea numerelor cu  $n$  cifre formate cu elementele mulțimii  $\{0, 2, 9\}$  se utilizează un algoritm backtracking care, pentru  $n=2$ , generează, în ordine, numerele 20, 22, 29, 90, 92, 99.  
Dacă  $n=4$  și se utilizează același algoritm, care este numărul generat imediat după numărul 2009? (4p.)
- a. 2002                      b. 2020                      c. 2090                      d. 2010

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Subprogramul **afis** este definit alăturat.  
Ce se afișează ca urmare a apelului **afis(8);**? (4p.)
- ```
void afis (int n)
{
    cout<<n; | printf("%d",n);
    for (int i=n/2;i>=1;i--)
        if(n%i==0)afis(i);
}
```
3. Scrieți programul C/C++ care citește de la tastatură două numere naturale nenule  $n$  și  $k$  ( $n \leq 100$ ,  $k \leq 100$ ) și afișează pe ecran, separați prin câte un spațiu, în **ordine descrescătoare**, cei mai mici  $k$  multipli naturali nenuli ai numărului  $n$ .  
**Exemplu:** pentru  $n=6$  și  $k=5$  se afișează 30 24 18 12 6. (6p.)
4. a) Scrieți definiția completă a subprogramului **sterge**, care primește prin cei 4 parametri  $v, n, i, j$ :
- $v$ , un tablou unidimensional cu maximum 100 de elemente întregi din intervalul  $[-1000, 1000]$
  - $n$ , un număr natural reprezentând numărul de elemente din tabloul  $v$
  - $i$  și  $j$  două valori naturale cu  $1 \leq i \leq j \leq n$
- și elimină din tabloul  $v$  elementele  $v_i, v_{i+1}, \dots, v_j$  actualizând valoarea parametrului  $n$ .  
Tabloul modificat este furnizat tot prin parametrul  $v$ . (6p.)
- b) Fișierul text **NUMERE.IN** conține pe prima linie un număr natural nenul  $n$  ( $1 \leq n \leq 100$ ) și pe următoarea linie  $n$  numere întregi din intervalul  $[-1000, 1000]$ , separate prin câte un spațiu.  
Scrieți un program C/C++ care citește din fișierul **NUMERE.IN** numărul natural  $n$ , construiește în memorie un tablou unidimensional  $v$  cu cele  $n$  numere întregi aflate pe linia a doua în fișier și utilizează apeluri utile ale subprogramului **sterge** pentru a elimina din tablou un număr minim de elemente astfel încât să nu existe două elemente alăturate cu aceeași valoare. Elementele tabloului obținut se afișează pe ecran, separate prin câte un spațiu.  
**Exemplu:** Dacă fișierul **NUMERE.IN** are conținutul:  
12  
10 10 2 2 19 9 9 9 9 15 15 15 atunci se afișează 10 2 19 9 15. (10p.)

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Pentru generarea în ordine crescătoare a numerelor cu  $n$  cifre formate cu elementele mulțimii  $\{0, 2, 8\}$  se utilizează un algoritm backtracking care, pentru  $n=2$ , generează, în ordine, numerele 20, 22, 28, 80, 82, 88.  
Dacă  $n=4$  și se utilizează același algoritm, precizați câte numere generate sunt divizibile cu 100? (4p.)
- a. 8                                      b. 90                                      c. 6                                      d. 10

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Subprogramul `scrie` este definit alăturat. Ce se afișează ca urmare a apelului `scrie(2,6);`? (6p.)
- ```
void scrie (int x,int y)
{
    cout<<x<<y; | printf("%d%d",x,y);
    if(x<y)
    {
        scrie(x+1,y-1);
        cout<<(x+y)/2;|printf("%d",(x+y)/2);
    }
}
```
3. Scrieți definiția completă a subprogramului `nreal` cu doi parametri  $x$  și  $y$ , numere naturale din intervalul  $[1;1000]$  ce returnează un număr real cu proprietatea că partea sa întreagă este egală cu  $x$ , iar numărul format din zecimalele sale, în aceeași ordine, este egal cu  $y$ .  
**Exemplu:** pentru  $x=12$  și  $y=543$ , subprogramul returnează valoarea 12.543. (10p.)
4. Fișierul text **NUMERE.IN** conține pe prima linie un număr natural nenul  $n$  ( $2 \leq n \leq 100$ ) și pe următoarea linie  $n$  numere reale pozitive, în ordine strict crescătoare, separate prin câte un spațiu.
- a) Scrieți un program C/C++ care, utilizând un algoritm eficient din punct de vedere al memoriei utilizate, determină și afișează pe ecran cel mai mare număr natural  $x$  cu proprietatea că în orice interval deschis având drept capete oricare două dintre cele  $n$  numere aflate pe linia a doua în fișierul **NUMERE.IN** se găsesc cel puțin  $x$  numere întregi.  
**Exemplu:** dacă fișierul **NUMERE.IN** are conținutul:
- ```
6
3.5  5.1  9.2  16  20.33 100
```
- atunci se afișează 2
- Explicație: în oricare dintre intervalele  $(3.5, 5.1)$ ,  $(3.5, 9.2)$ ,  $(3.5, 16)$ ,  $(3.5, 20.33)$ ,  $(3.5, 100)$ ,  $(5.1, 9.2)$ ,  $(5.1, 16)$ ,  $(5.1, 20.33)$ ,  $(5.1, 100)$ ,  $(9.2, 16)$ ,  $(9.2, 20.33)$ ,  $(9.2, 100)$ ,  $(16, 20.33)$ ,  $(16, 100)$ ,  $(20, 33, 100)$  există cel puțin două numere întregi.
- b) Descrieți în limbaj natural metoda utilizată și explicați în ce constă eficiența ei. (4p.)



**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. În câte dintre permutările elementelor mulțimii  $\{ 'I', 'N', 'F', 'O' \}$  vocalele apar pe poziții consecutive? (4p.)
- a. 24                      b. 6                      c. 12                      d. 4

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Ce se afișează ca urmare a apelului `p(123)`; dacă subprogramul `p` are definiția alăturată? (6p.)
- ```
void p (int x)
{cout<<x; | printf("%d",x);
 if(x!=0){p(x/10);
  cout<<x%10; | printf("%d",x%10);}}
```

3. Scrieți definiția completă a subprogramului `multipli`, cu trei parametri `a, b, c` ( $a \leq b$ ), numere naturale din intervalul  $[1, 10000]$  ce returnează numărul multiplilor lui `c` din intervalul  $[a; b]$ .

**Exemplu:** pentru  $a=10, b=27, c=5$  subprogramul returnează valoarea 4. (10p.)

4. Se consideră două tablouri unidimensionale **A** și **B** cu elemente numere naturale din intervalul  $[1, 10000]$ . Spunem că tabloul **A** "**se poate reduce**" la tabloul **B** dacă există o împărțire a tabloului **A** în secvențe disjuncte de elemente aflate pe poziții consecutive în tabloul **A** astfel încât prin înlocuirea secvențelor cu suma elementelor din secvență să se obțină, în ordine, elementele tabloului **B**.

De exemplu tabloul

|   |    |   |   |   |   |    |   |   |   |    |   |   |
|---|----|---|---|---|---|----|---|---|---|----|---|---|
| A | 7  | 3 | 4 | 1 | 6 | 4  | 6 | 9 | 7 | 1  | 8 | 7 |
| B | 14 |   |   | 7 |   | 26 |   |   |   | 16 |   |   |

se poate reduce la tabloul

Fișierul text **NUMERE.IN** conține pe prima linie două numere naturale nenule  $n$  și  $m$  ( $1 \leq m \leq n \leq 100$ ), pe linia a doua  $n$  numere naturale din intervalul  $[1; 10000]$  și pe linia a treia alte  $m$  numere naturale din intervalul  $[1; 10000]$ . Pe fiecare linie numerele sunt separate prin câte un spațiu.

**a)** Scrieți un program C/C++ care citește toate numerele din fișierul **NUMERE.IN** și verifică, utilizând un algoritm eficient din punctul de vedere al timpului de executare, dacă tabloul construit cu cele  $n$  numere aflate pe linia a doua în fișier se poate reduce la tabloul construit cu cele  $m$  numere aflate pe linia a treia în fișier. Programul afișează pe ecran mesajul **DA** în caz afirmativ și mesajul **NU** în caz negativ. (6p.)

**b)** Descrieți în limbaj natural metoda utilizată și explicați în ce constă eficiența ei. (4p.)

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Pentru generarea numerelor cu  $n$  cifre formate cu elementele mulțimii  $\{0, 4, 8\}$  se utilizează un algoritm backtracking care, pentru  $n=2$ , generează, în ordine, numerele 40, 44, 48, 80, 84, 88.  
Dacă  $n=4$  și se utilizează același algoritm, care este numărul generat imediat după numărul 4008 ? (4p.)
- a. 4040                      b. 4004                      c. 4080                      d. 8004

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Subprogramul `f` este definit alăturat.  
Ce se afișează ca urmare a apelului `f(1,3);`? (6p.)
- ```
void f (int x,int y)
{int i;
 for (i=x;i<=y;i++)
 {
  cout<<i; | printf("%d",i);
  f(i+1,y);
 }
}
```
3. Scrieți definiția completă a subprogramului `suma`, care primește prin cei 4 parametri,  $v, n, i, j$ :
- $v$ , un tablou unidimensional cu maximum 100 de elemente întregi din intervalul  $[-1000, 1000]$ , numerotate de la 1 la  $n$ ;
  - $n$ , un număr natural reprezentând numărul de elemente din tabloul  $v$ ;
  - $i$  și  $j$ , două valori naturale cu  $1 \leq i \leq j \leq 100$
- și returnează suma elementelor  $v_1, \dots, v_{i-1}, v_{j+1}, \dots, v_n$  din tabloul  $v$ . (10p.)
4. Fișierul text **NUMERE.IN** conține pe prima linie un număr natural nenul  $n$  ( $1 \leq n \leq 100$ ) și pe următoarea linie  $n$  numere reale pozitive **ordonate crescător**, separate prin câte un spațiu.
- a) Scrieți un program C/C++ care citește din fișierul **NUMERE.IN** numărul natural  $n$ , și determină, utilizând un algoritm eficient din punct de vedere al timpului de executare și al memoriei utilizate, numărul **minim** de intervale închise de forma  $[x, x+1]$ , cu  $x$  număr natural, a căror reuniune include toate numerele reale din fișier.

**Exemplu:** Dacă fișierul **NUMERE.IN** are conținutul:

6

2.3   2.3   2.8   5.7   5.7   6.3

atunci se afișează 3 (intervalele  $[2,3]$ ,  $[5,6]$ ,  $[6,7]$  sunt cele 3 intervale de forma cerută care conțin numere din șir). (6p.)

b) Descrieți în limbaj natural metoda utilizată și explicați în ce constă eficiența ei. (4p.)

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Având la dispoziție cifrele 0, 1 și 2 putem genera, în ordine crescătoare, numere care au suma cifrelor egală cu 2 astfel încât primele 6 numere generate sunt, în această ordine: 2, 11, 20, 101, 110, 200. Folosind același algoritm se generează numere cu cifrele 0, 1, 2 și 3 care au suma cifrelor egală cu 4. Care va fi al 7-lea număr din această generare ? **(4p.)**
- a. 103                      b. 301                      c. 220                      d. 130

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Se consideră subprogramul recursiv `f1` definit alăturat. Ce se va afișa în urma apelului `f1(5);`? **(6p.)**
- ```
void f1(int x)
{ if (x<=9)
  { cout<<x+1; | printf("%d",x+1);
    f1(x+2);
    cout<<x+3; | printf("%d",x+3);
  }
}
```
3. Scrieți definiția completă a subprogramului `suma` care primește ca parametru un tablou unidimensional `x` cu cel mult 100 de elemente, numere reale, un număr natural `n` ce reprezintă numărul efectiv de elemente ale tabloului `x` ( $n \leq 100$ ), și un număr natural `m` ( $n \geq m$ ). Subprogramul returnează suma obținută din cele mai mici `m` elemente ale tabloului `x`. **(10p.)**
4. În fișierul `numere.txt` se află memorate, pe prima linie un număr natural `n` ( $1 \leq n \leq 100$ ), iar pe fiecare dintre următoarele `n` linii, câte două numere întregi `x, y` ( $-100 \leq x \leq y \leq 100$ ), reprezentând capetele câte unui segment `[x, y]` desenat pe axa `Ox` de coordonate.
- a) Scrieți în limbajul C/C++ un program eficient din punct de vedere al timpului de executare și al spațiului de memorare, care citește din fișier datele existente, determină segmentul rezultat în urma intersecției tuturor celor `n` segmente date și afișează pe ecran două numere despărțite printr-un spațiu ce reprezintă capetele segmentului cerut. Dacă segmentele nu au nici un punct comun se va afișa pe ecran valoarea 0. **(6p.)**
- b) Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia. **(4p.)**

**Exemplu:** dacă fișierul `numere.txt` are conținutul alăturat, se va afișa

|          |       |
|----------|-------|
| pe ecran | 5     |
| 3 5      | -7 10 |
|          | 3 20  |
|          | -5 5  |
|          | 0 12  |
|          | -8 30 |

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. În vederea participării la un concurs, elevii de la liceul sportiv au dat o probă de selecție, în urma căreia primii 6 au obținut punctaje egale. În câte moduri poate fi formată echipa selecționată știind că poate avea doar 4 membri, aleși dintre cei 6, și că ordinea acestora în cadrul echipei nu contează? **(4p.)**
- a. 24                                      b. 30                                      c. 15                                      d. 4

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Subprogramul `afis` este definit alăturat. Ce se va afișa în urma apelului `afis(17);`? **(6p.)**
- ```
void afis(int x)
{
    if (x>3)
    {
        cout<<x-1; | printf("%d",x-1);
        afis(x/3);
        cout<<x+1; | printf("%d",x+1);
    }
}
```
3. Scrieți definiția completă a subprogramului `nr_prim` care are ca parametru un număr natural `x` (`x<10000`) și returnează cel mai mic număr prim, strict mai mare decât `x`.  
**Exemplu:** pentru `x=25` subprogramul returnează numărul 29, iar pentru `x=17` valoarea returnată va fi 19. **(10p.)**
4. În fișierul `numere.txt` sunt memorate pe mai multe linii, numere întregi (cel mult 100), numerele de pe aceeași linie fiind despărțite prin câte un spațiu, fiecare număr având cel mult 9 cifre. Să se determine cele mai mici două valori distincte, fiecare având **exact** două cifre, memorate în fișier și să se afișeze pe ecran aceste valori, despărțite printr-un spațiu. Dacă în fișier nu se află două astfel de valori, pe ecran se va afișa valoarea 0.
- a) Descrieți în limbaj natural o metodă de rezolvare eficientă din punct de vedere al gestionării memoriei și al timpului de executare. **(4p.)**
- b) Scrieți programul C/C++ corespunzător metodei descrise la punctul a. **(6p.)**
- Exemplu:** dacă fișierul `numere.txt` are conținutul alăturat, se
- |    |              |
|----|--------------|
| 5  | 10           |
| 3  | -77 20       |
| 50 | 5 0 12 18 30 |
- va afișa pe ecran, nu neapărat în această ordine:  
-77 10



**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Folosind un algoritm de generare putem obține numere naturale de  $k$  cifre care au suma cifrelor egală cu un număr natural  $s$ . Astfel, pentru valorile  $k=2$  și  $s=6$  se generează, în ordine, numerele: 15, 24, 33, 42, 51, 60. Care va fi al treilea număr generat pentru  $k=4$  și  $s=5$ ? (4p.)
- a. 1301                      b. 1022                      c. 2201                      d. 1031

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Subprogramul `re` este definit alăturat. Ce valoarea are `re(1)`? Dar `re(14)`? (6p.)
- ```
int re(int i)
{
    if (i<9) return 3+re(i+2);
    else
        if (i==9) return -2;
        else return 1+re(i-1);
}
```
3. Subprogramul `sum3` primește prin parametrul `x` un tablou unidimensional, cu cel mult 100 de elemente, numere întregi cu cel mult 4 cifre fiecare, iar prin parametrul `n` un număr natural ce reprezintă numărul efectiv de elemente ale tabloului `x` ( $n \leq 100$ ). Scrieți definiția completă a subprogramului `sum3` care returnează suma elementelor tabloului care sunt divizibile cu 3. Dacă tabloul nu conține nicio valoare divizibilă cu 3, subprogramul va returna 0. (6p.)
4. Pe prima linie a fișierului `numere.txt` se află un număr natural  $n$  ( $n \leq 100$ ), iar pe următoarele  $n$  linii, câte  $n$  numere naturale despărțite prin câte un spațiu, fiecare având cel mult 9 cifre. Printre aceste numere se află cel puțin unul cu 3 cifre și cel puțin unul cu 4 cifre.
- a) Scrieți în limbajul C/C++, un algoritm eficient din punct de vedere al gestionării memoriei care citește din fișier datele existente și determină și afișează pe ecran, separate printr-un spațiu, două numere din fișier,  $x$  și  $y$ , unde  $x$  este cel mai mare număr de trei cifre, iar  $y$  este acel număr pentru care  $|x-y|$  are valoare minimă. Dacă sunt mai multe valori pentru  $y$  care respectă condiția impusă se va afișa numai una dintre ele. (10p.)
- b) Explicați în limbaj natural metoda utilizată justificând eficiența acesteia. (4p.)

**Exemplu:** dacă fișierul `numere.txt` are

|                                   |                    |
|-----------------------------------|--------------------|
| conținutul alăturat, se va afișa: | 5                  |
|                                   | 112 333 1 500 1100 |
|                                   | 1 95 7 97 12       |
|                                   | 45 800 0 7 89      |
|                                   | 1 5 17 197 102     |
|                                   | 45 86 0 7 9        |

800 1100

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Completarea unui bilet de LOTO presupune colorarea a 6 numere dintre cele 49, înscrise pe bilet. O situație statistică pe o anumită perioadă de timp arată că cele mai frecvente numere care au fost extrase la LOTO sunt: 2, 20, 18, 38, 36, 42, 46, 48. Câte bilete de 6 numere se pot completa folosind doar aceste valori, știind că numărul 42 va fi colorat pe fiecare bilet? (4p.)
- a. 21                      b. 6!                      c. 42                      d. 56

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Se consideră definit subprogramul `f`.  
Scrieți două valori naturale, `x1` și `x2` (`x1 ≠ x2`, `x1 < 12` și `x2 < 12`) pentru care `f(x1) = f(x2)`. (6p.)
- ```
int f(int i)
{
    if (i > 12) return 1;
    else return 1+f(i+2);
}
```
3. a) Subprogramul `max` primește ca parametru un tablou unidimensional `x` cu cel mult 100 de elemente numere întregi, care sunt, în ordine, termenii unei progresii aritmetice și un număr natural `n`, care reprezintă dimensiunea tabloului. Scrieți definiția completă a subprogramului `max` care returnează cel mai mare termen al progresiei aritmetice. Alegeți un algoritm de rezolvare eficient din punct de vedere al timpului de executare. (6p.)
- b) Explicați în limbaj natural metoda utilizată justificând eficiența acesteia. (4p.)
- c) Pe prima linie a fișierului `numere.txt` se află un număr natural `n` (`n ≤ 100`), iar pe următoarele `n` linii, câte `n` numere întregi cu cel mult 4 cifre fiecare. Scrieți programul C/C++ care citește din fișier datele existente, determină liniile din fișier pe care s-au memorat în ordine termenii unei progresii aritmetice și afișează pe ecran, folosind apeluri ale subprogramului `max` cel mai mare număr (diferit de cel situat pe prima linie) din fișier, care în plus este termenul unei progresii aritmetice. (10p.)

**Exemplu:** dacă fișierul `numere.txt` are conținutul alăturat, se va afișa 50, deoarece progresiile aritmetice sunt:

(-9 -7 -5 -3 -1),  
(50 40 30 20 10) și  
(18 17 16 15 14)

```
5
5 7 3 1 9
-9 -7 -5 -3 -1
2 5 8 14 11
50 40 30 20 10
18 17 16 15 14
```

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Pentru generarea tuturor mulțimilor de câte 5 cifre, având la dispoziție cifrele de la 1 la 9, se poate utiliza un algoritm echivalent cu algoritmul de generare a: **(4p.)**
- a. permutărilor de 5 elemente      b. submulțimilor mulțimii {1,2,3,4,5,6,7,8,9}
- c. combinațiilor de 9 elemente luate câte 5      d. aranjamentelor de 9 elemente luate câte 5

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Subprogramul `f` este definit alăturat. Ce se va afișa în urma executării secvenței de mai jos, în care variabilele `a` și `b` sunt de tip întreg? **(6p.)**
- ```
void f(int &a,int b)
{
    a=a-5;b=a-2;
    cout<<a<<b; | printf("%d%d",a,b);
}
```
3. Fișierul text `numere.in` conține pe prima linie un număr natural `n` ( $0 < n < 1000$ ), iar pe a doua linie `n` numere naturale cu cel mult 9 cifre fiecare, despărțite prin câte un spațiu. Scrieți un program C/C++ care citește toate numerele din fișier și afișează pe ecran, separate prin câte un spațiu, numerele de pe a doua linie a fișierului, care încep și se termină cu aceeași cifră. **(10p.)**

**Exemplu:** dacă fișierul `numere.in` are conținutul alăturat, se numerele ce se vor afișa sunt: 55 3 101 7 2782

```
9
55 107 3 101 92 7 208 2782
80
```

4. a) Scrieți definiția completă a unui subprogram `sum` care primește prin parametrul `x` un număr natural de cel mult 4 cifre și returnează suma divizorilor numărului `x`, diferiți de 1 și de el însuși.

**Exemplu:** dacă `x=10` se va returna valoarea 7 ( $7=2+5$ ). **(4p.)**

b) Scrieți programul C/C++ care citește de la tastatură un număr natural `n` ( $0 < n < 100$ ), apoi `n` numere naturale (cu cel mult 4 cifre fiecare). Programul determină, folosind apeluri utile ale subprogramului `sum`, pentru fiecare număr natural citit, suma divizorilor săi proprii și afișează pe ecran sumele determinate, în ordinea crescătoare a valorilor lor, separate prin câte un spațiu. **(6p.)**

**Exemplu:** dacă `n=5` și numerele citite sunt 10 2 33 6 11

valorile afișate pe ecran vor fi: 0 0 5 7 14

deoarece suma divizorilor lui 10 este 7, suma divizorilor lui 2 este 0, suma divizorilor lui 33 este 14, suma divizorilor lui 6 este 5, suma divizorilor lui 11 este 0.

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Subprogramul `scif` returnează suma cifrelor unui număr natural transmis ca parametru. Care este valoarea expresiei `scif(scif(518)+scif(518))`? **(4p.)**
- a. 14                                      b. 10                                      c. 28                                      d. 1

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Utilizăm metoda backtracking pentru generarea tuturor modalităților de a scrie numărul 9 ca sumă a cel puțin două numere naturale nenule distincte. Termenii fiecărei sume sunt în ordine strict crescătoare. Soluțiile se generează în ordine: 1+2+6, 1+3+5, 1+8, 2+3+4, 2+7, 3+6 și 4+5. Se aplică exact aceeași metodă pentru scrierea lui 12. Scrieți, în ordinea generării, toate soluțiile de forma 2+... **(6p.)**
3. Subprogramul `cifra` primește prin intermediul parametrului `a` un număr natural cu cel mult 4 cifre și returnează ultima cifră pară a sa. Dacă numărul nu conține cifre pare, subprogramul returnează valoarea -1. De exemplu, dacă `a=8345`, subprogramul va returna 4.

**a) Să se scrie definiția completă a subprogramului `cifra`. **(10p.)****

**b) Pe prima linie a fișierului `bac.in` se află un număr natural nenul `n` ( $n \leq 15000$ ), iar pe a doua linie a fișierului se află un șir de `n` numere naturale, despărțite prin câte un spațiu, fiecare număr fiind format din cel mult 4 cifre.**

Scrieți un program C/C++ care citește numerele din fișier și afișează pe ecran, folosind apeluri utile ale subprogramului `cifra`, cel mai mare număr care se poate forma cu ultimele cifre pare ale fiecărui element, dacă acestea există. Alegeți o metodă de rezolvare eficientă ca timp de executare. Dacă toate numerele de pe a doua linie a fișierului au numai cifre impare, programul va afișa mesajul **NU EXISTA**.

**Exemplu:** dacă fișierul `bac.in` are conținutul 

|                        |  |
|------------------------|--|
| 7                      |  |
| 369 113 2 0 33 1354 42 |  |

  
alăturat, pe ecran se va afișa: **64220** **(6p.)**

**c) Descrieți succint în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența ei (3 - 4 rânduri). **(4p.)****

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Se utilizează un algoritm pentru a genera în ordine lexicografică inversă toate permutările mulțimii  $\{1,2,3,4,5\}$ . Primele patru permutări generate sunt: 54321, 54312, 54231, 54213. A cincea permutare este: **(4p.)**
- a. 53421                      b. 54321                      c. 54132                      d. 54123

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Considerăm subprogramul `f` definit alăturat. Ce valoare are `f(7,11)`? Dar `f(11,7)`? **(6p.)**
- ```
int f(int x,int y)
{
    if(x<=y) return x-y;
    return f(y-x,x-1)+3;
}
```
3. Scrieți un program C/C++ care citește numerele naturale nenule  $n$  și  $k$  ( $k \leq n \leq 100$ ) și un tablou unidimensional cu  $n$  elemente numere întregi, fiecare având cel mult 4 cifre. Programul modifică tabloul, permutând circular, cu  $k$  poziții spre stânga, elementele acestuia și afișează pe ecran, separate prin câte un spațiu, elementele tabloului obținut.
- Exemplu:** dacă  $n=4$ ,  $k=3$  și tabloul  $v=(1,2,3,4)$ , atunci se vor afișa în ordine elementele: 4 1 2 3. **(10p.)**
4. **a)** Scrieți doar antetul subprogramului `nrdiv`, care primește prin intermediul parametrului  $x$  un număr natural nenul cu cel mult 4 cifre, și returnează numărul de divizori primi ai lui  $x$ . **(4p.)**
- b)** Pe prima linie a fișierului `bac.in` se află un număr natural nenul  $n$  ( $n \leq 1000$ ), iar pe a doua linie a fișierului se află un șir format din  $n$  numere naturale nenule, despărțite prin câte un spațiu, fiecare număr fiind format din cel mult 4 cifre. Scrieți un program C/C++ care citește numerele din fișier și care afișează pe ecran, folosind apeluri utile ale subprogramului `nrdiv`, prima și ultima valoare din șirul celor  $n$  numere citite, care au un număr par de divizori primi. Numerele afișate vor fi separate printr-un spațiu.

**Exemplu:** dacă fișierul `bac.in` are conținutul alăturat, pe ecran se va afișa: 20 10 **(6p.)**

```
7
30 105 20 140 7 10 5
```

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Utilizăm metoda backtracking pentru generarea tuturor modalităților de a scrie numărul 9 ca sumă a cel puțin două numere naturale nenule distincte. Termenii fiecărei sume sunt în ordine strict crescătoare. Soluțiile se generează în ordinea: 1+2+6, 1+3+5, 1+8, 2+3+4, 2+7, 3+6 și 4+5. Se aplică exact aceeași metodă pentru scrierea lui 8. Câte soluții vor fi generate? **(4p.)**
- a. 3                      b. 4                      c. 6                      d. 5

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Pentru definiția alăturată a subprogramului `sc`, stabiliți ce valoare are `sc(10)`. Dar `sc(901324)`? **(6p.)**
- ```
int sc(long x)
{
    if(x<10) return x;
    return sc(x/10)+x%10;
}
```
3. Scrieți în limbajul C/C++ definiția completă a subprogramului `Del` care are doi parametri: `x`, un număr întreg de cel mult 9 cifre, și `y`, un număr natural nenul de o cifră. Subprogramul determină eliminarea tuturor cifrelor lui `x` mai mari strict decât `y` și furnizează numărul obținut tot prin intermediul parametrului `x`. Dacă toate cifrele lui `x` sunt mai mari strict decât `y`, atunci `x` va primi valoarea -1. **(10p.)**
- Exemplu:** dacă `x=37659` și `y=6`, după apel `x=365`, iar `y=6`.
4. Se consideră subprogramul `inter`, cu doi parametri: `x` și `y` (numere întregi formate din cel mult patru cifre fiecare); subprogramul interschimbă valorile a două variabile transmise prin intermediul parametrilor `x` și `y`. **(4p.)**
- a) Scrieți în limbajul C/C++ definiția completă a subprogramului `inter`. **(4p.)**
- b) Pe prima linie a fișierului `bac.in` se află un număr natural nenul  $n \leq 1000$ , iar pe a doua linie a fișierului se află un șir de `n` numere naturale nenule, despărțite prin câte un spațiu, fiecare număr fiind format din cel mult 4 cifre. Scrieți un program C/C++ care afișează pe ecran, în ordine crescătoare, numerele aflate pe a doua linie a fișierului. Numerele vor fi afișate pe o singură linie, iar între două numere se va lăsa un spațiu. Se vor folosi apeluri utile ale subprogramului `inter`. **(6p.)**

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Utilizăm metoda backtracking pentru generarea tuturor modalităților de a scrie numărul 6 ca sumă a cel puțin două numere naturale nenule. Termenii fiecărei sume sunt în ordine crescătoare. Soluțiile se generează în ordinea: 1+1+1+1+1+1, 1+1+1+1+2, 1+1+1+1+3, 1+1+2+2, 1+1+4, 1+2+3, 1+5, 2+2+2, 2+4 și 3+3. Se aplică exact aceeași metodă pentru scrierea lui 9. Care este penultima soluție?

**(4p.)**

- a. 3+3+3                      b. 3+6                      c. 4+5                      d. 2+7

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Pentru definiția alăturată a subprogramului `f`, ce valoare are `f(3)`? Dar `f(8)`? **(6p.)**
- ```
int f(int x)
{
    if(x<=4) return x*x-3;
    return f(x-3)+4;
}
```

3. Pe prima linie a fișierului `bac.in` se află un număr natural nenul `n` ( $n \leq 1000$ ), iar pe a doua linie a fișierului se află un șir format din `n` numere naturale, despărțite prin câte un spațiu, fiecare număr fiind format din cel mult 4 cifre. Scrieți un program C/C++ care citește valorile din fișier și care afișează pe ecran mesajul **DA** dacă toate elementele pare din șir sunt în ordine crescătoare și mesajul **NU** în caz contrar.

**Exemplu:** dacă fișierul `bac.in` are conținutul

	8
alăturat, pe ecran se va afișa: <b>DA</b>	<b>(10p.)</b> <u>10</u> <u>113</u> <u>12</u> <u>33</u> <u>12</u> <u>42</u> <u>1354</u> <u>421</u>

4. Se consideră subprogramul `pr`, care primește prin intermediul parametrului `a` un număr natural nenul cu cel mult 9 cifre și returnează 1 dacă numărul este prim și 0 în caz contrar.

**a) Scrieți numai antetul subprogramului `pr`. (4p.)**

**b) Considerăm un număr natural nenul  $n > 99$  cu cel mult 9 cifre. Din `n` se obține un șir de valori prin eliminarea succesivă a ultimei cifre, apoi a ultimelor două cifre, apoi a ultimelor trei cifre etc., până ce se obține un număr de două cifre, ca în exemplu. Să se realizeze un program C/C++ care citește de la tastatură numărul `n` și care, folosind apeluri utile ale subprogramului `pr`, afișează pe ecran, separate prin câte un spațiu, doar valorile prime din șirul numerelor obținute din `n`, prin procedeul descris mai sus.**

**Exemplu:** pentru `n=193124` se obține șirul de valori 19312, 1931, 193, 19. din care se vor afișa pe ecran doar valorile 1931 193 19 (nu neapărat în această ordine). **(6p.)**

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Utilizăm metoda backtracking pentru generarea tuturor modalităților de a scrie numărul 6 ca sumă a cel puțin două numere naturale nenule. Termenii fiecărei sume sunt în ordine crescătoare. Soluțiile se generează în ordinea: 1+1+1+1+1+1, 1+1+1+1+2, 1+1+1+1+3, 1+1+2+2, 1+1+4, 1+2+3, 1+5, 2+2+2, 2+4 și 3+3. Se aplică exact aceeași metodă pentru scrierea lui 9. Câte soluții de forma 2+... vor fi generate? **(4p.)**
- a. 2                      b. 3                      c. 4                      d. 5

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. În secvența alăturată variabilele *i*, *j* și *aux* sunt întregi, iar *v* memorează un tablou unidimensional ale cărui elemente au următoarele valori: *v*[1]=2, *v*[2]=13, *v*[3]=1, *v*[4]=5, *v*[5]=-4. Care este numărul de interschimbări care se efectuează în urma executării secvenței alăturate? Dați un exemplu de alte 5 valori pentru elementele tabloului astfel încât, în urma executării secvenței alăturate, să nu se efectueze nicio interschimbare. **(6p.)**
- ```
for(i=1;i<5;i++)
for(j=5;j>i;j--)
if(v[i]<v[j])
{
    aux=v[i];
    v[i]=v[j];
    v[j]=aux;
}
```
3. a) Pe prima linie a fișierului **bac.in** se află un număr natural nenul *n* ( $n \leq 1000$ ), iar pe a doua linie a fișierului se află un șir format din *n* numere naturale, despărțite prin câte un spațiu, fiecare număr fiind format din cel mult 4 cifre. Scrieți un program C/C++ care citește numerele din fișier și care afișează pe ecran mesajul **DA** dacă elementele pare în șir sunt în ordine crescătoare, iar cele impare sunt în ordine descrescătoare și mesajul **NU** în caz contrar. Alegeți un algoritm eficient ca timp de executare și spațiu de memorie utilizat. **(6p.)**
- b) Descrieți succint, în limbaj natural, metoda utilizată, justificând eficiența acesteia. **(4p.)**
- Exemplu:** dacă fișierul **bac.in** are conținutul
- |    |                            |
|----|----------------------------|
| 10 |                            |
| 10 | 1133 12 331 12 42 1354 221 |
| 13 | 13                         |
- alăturat, pe ecran se va afișa: **DA**
4. a) Scrieți definiția completă a subprogramului **sdiv** care primește prin intermediul parametrului *y* un număr natural cu cel mult 6 cifre și returnează suma tuturor divizorilor numărului *y*. **(5p.)**
- b) Scrieți un program C/C++ care citește de la tastatură un număr natural *n* ( $n < 10000$ ) și care, folosind apeluri ale subprogramului **sdiv** verifică dacă suma divizorilor lui *n* este un număr prim. În caz afirmativ, programul va afișa pe ecran mesajul **Da** și în caz contrar va afișa mesajul **NU**.
- Exemplu:** dacă *n*=206, atunci programul va afișa: **NU** (1+2+103+206=312, iar 312 nu este un număr prim). **(5p.)**



**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Cea mai mare dintre valorile strict negative memorate într-un tablou unidimensional NU există dacă: **(4p.)**
- a. în tablou sunt memorate numai valori strict negative
  - b. în tablou există cel puțin o valoare strict negativă
  - c. în tablou sunt memorate valori nenule pozitive și negative
  - d. în tablou sunt memorate numai valori strict pozitive

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Știind că  $p$  este un vector cu 3 componente întregi (vector declarat global), stabiliți cu ce trebuie înlocuite  $\alpha$  și  $\beta$  în definiția subprogramului  $G$  alăturat astfel încât în urma apelului  $G(0)$  să se afișeze toate numerele de 3 cifre nenule, fiecare număr fiind afișat o singură dată. **(6p.)**
- ```
void G(int k)
{
    int i;
    for(i=1; i<= $\alpha$ ; i++)
    {
        p[k]=i;
        if( $\beta$ )G(k+1);
        else
            cout<<p[0]<<p[1]<<p[2]<<endl; |
            printf("%d%d%d\n",p[0],p[1],p[2]);
    }
}
```

3. Scrieți un subprogram **DIST**, cu doi parametri, care primește prin intermediul parametrului  $a$  un tablou unidimensional cu cel mult 100 de elemente, numere naturale de cel mult 4 cifre fiecare, și prin intermediul parametrului  $n$  un număr natural nenul,  $n < 100$ , ce reprezintă numărul de elemente din tablou. Subprogramul returnează valoarea 1 dacă toate elementele tabloului  $a$  sunt distincte și dacă diferența absolută a oricăror două elemente vecine din tablou este diferită de 1, altfel returnând valoarea 0. **(10p.)**

4. Fișierul text **numere.txt** conține pe prima linie un număr natural  $n$  ( $0 < n < 100000$ ), iar pe a doua linie un șir format din  $n$  cifre, separate prin câte un spațiu.

a) Scrieți un program **C/C++** care determină în mod eficient din punct de vedere al timpului de executare, cea mai mare cifră dintre cele situate pe a doua linie a fișierului, precum și numărul de apariții ale acesteia. Cele două numere vor fi afișate pe o singură linie a ecranului, separate printr-un spațiu.

**Exemplu:** dacă fișierul **numere.txt** are următorul conținut:

7

3 5 2 1 5 3 1

atunci pe ecran se va afișa: 5 2.

**(6p.)**

b) Descrieți succint, în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența ei (3 – 4 rânduri). **(4p.)**

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Utilizând metoda backtracking se generează toate permutările mulțimii  $\{1, 2, 3, 4\}$ . Dacă primele trei permutări generate sunt, în această ordine: 1234, 1243, 1324 precizați care este permutarea generată imediat după 3412. **(4p.)**
- a. 3214                      b. 3413                      c. 4123                      d. 3421

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Subprogramul **f** are definiția alăturată. Ce valoare are **f(7)**? Dar **f(100)**? **(6p.)**
- ```
int f(int x)
{
    if(x%6==0) return x;
    else return f(x-1);
}
```
3. Scrieți definiția completă a unui subprogram **P**, cu trei parametri, care primește prin intermediul primului parametru, **a**, un tablou unidimensional de cel mult 100 de numere întregi, cu cel mult 4 cifre fiecare, prin intermediul celui de-al doilea parametru, **n**, numărul efectiv de elemente ale tabloului, iar prin parametrul **k**, un număr natural ( $k < n$ ) și returnează cea mai mare sumă cu **k** termeni care se poate obține adunând **k** elemente ale tabloului.
- Exemplu:** dacă  $n=6$  și  $k=4$ , iar șirul este format din elementele (5, 2, 5, 4, 1, 3), atunci la apel se va returna 17. **(10p.)**
4. Fișierul text **numere.txt** conține pe prima linie un număr natural **n** ( $0 < n < 100000$ ) iar pe a doua linie, separate prin câte un spațiu, **n** numere naturale formate din cel mult două cifre fiecare.
- a)** Scrieți un program C/C++ care determină în mod eficient, din punct de vedere al timpului de executare, numerele ce apar o singură dată în a doua linie a fișierului. Aceste numere vor fi afișate pe ecran în ordine crescătoare, separate prin câte un spațiu.
- Exemplu:** dacă fișierul **numere.txt** are următorul conținut:
- ```
7
3 5 2 1 5 23 1
```
- atunci pe ecran se va afișa: 2 3 23. **(6p.)**
- b)** Descrieți succint, în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența ei (3 – 4 rânduri). **(4p.)**

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Utilizând metoda backtracking se generează numerele formate din câte 3 cifre distincte din mulțimea  $\{1, 3, 5, 7\}$ . Dacă primele trei numere generate sunt, în această ordine: 135, 137, 153 care este cel de-al patrulea număr generat? **(4p.)**
- a. 315                      b. 173                      c. 157                      d. 357

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Subprogramul **f** are definiția alăturată. Ce valoare are **f(3)**? Dar **f(10)**? **(6p.)**
- ```
int f(int x)
{
    if(x==0) return 0;
    else return f(x-1)+2;
}
```

```
int f(int x)
{
    if(x==0) return 0;
    else return f(x-1)+2;
}
```
3. Scrieți definiția completă a unui subprogram **P** cu doi parametri, care primește prin intermediul primului parametru, **n**, un număr natural nenul ( $1 \leq n \leq 100$ ) și prin intermediul celui de-al doilea parametru, **a**, un tablou unidimensional cu elementele numerotate de la 1 la **n**, numere întregi de cel mult 4 cifre fiecare. Subprogramul returnează suma tuturor numerelor **impare** aflate pe poziții **pare** din tablou.
- Exemplu:** dacă **n=6**, iar șirul **a** este format din elementele (3,12,7,1,4,3), atunci la apel se va returna 4. **(10p.)**
4. Fișierul text **numere.txt** conține pe prima linie un număr natural **n** ( $0 < n < 100000$ ) iar pe a doua linie **n** cifre, separate prin câte un spațiu.
- a)** Scrieți un program C/C++ care determină în mod eficient, din punct de vedere al timpului de executare, cel mai mare număr ce se poate forma cu toate cifrele conținute de a doua linie a fișierului **numere.txt**. Numărul determinat se va afișa pe ecran.
- Exemplu:** dacă fișierul **numere.txt** are următorul conținut:
- ```
7
2 5 3 1 5 8 9
```
- atunci pe ecran se va afișa: 9855321. **(6p.)**
- b)** Descrieți succint, în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența ei (3 – 4 rânduri). **(4p.)**

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Care dintre următoarele variante reprezintă antetul corect al unui subprogram care primește prin parametrii  $x$  și  $y$  două numere întregi și furnizează prin parametrul  $m$  cea mai mică dintre cele două valori  $x$  și  $y$ ? (4p.)
- a. `int minim(int x,int y,int m)`      b. `void minim(int x,int y,int &m)`  
c. `int minim(int x,int y)`              d. `void minim(int& x,int& y,int m)`

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Subprogramul `f` are definiția alăturată. Ce valoare are `f(6,5)`? Dar `f(5,10)`? (6p.)
- ```
int f(int x,int y)
{
    if(x==y)return x;
    else if(x<y)return f(x+1,y-1);
    else return f(x-1,y);
}
```

```
int f(int x,int y)
{
    if(x==y)return x;
    else if(x<y)return f(x+1,y-1);
    else return f(x-1,y);
}
```
3. Scrieți definiția completă a unui subprogram `P`, cu doi parametri,  $a$  și  $b$ , numere naturale cu cel mult 4 cifre fiecare, care afișează pe ecran, separate prin câte un spațiu, numerele aflate în intervalul închis determinat de valorile  $a$  și  $b$ , care sunt pătratele unor numere prime.  
**Exemplu:** pentru  $a=40$  și  $b=1$  se vor afișa valorile: 4 9 25 (nu neapărat în această ordine). (10p.)
4. Fișierul text `numere.txt` conține pe prima linie un număr natural  $n$  ( $0 < n < 100000$ ), iar pe a doua linie  $n$  numere naturale, formate din cel mult 4 cifre, separate prin câte un spațiu.  
a) Scrieți un program C/C++ care determină în mod eficient, din punct de vedere al timpului de executare, cifrele ce apar în scrierea numerelor situate pe a doua linie a fișierului. Programul va afișa pe ecran aceste cifre în ordine crescătoare, separate prin câte un spațiu.  
**Exemplu:** dacă fișierul `numere.txt` are următorul conținut:
- ```
7
243 32 545 74 12 1344 90
```
- atunci pe ecran se va afișa: 0 1 2 3 4 5 7 9 (6p.)
- b) Descrieți succint, în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența ei (3 – 4 rânduri). (4p.)

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Utilizând metoda backtracking se generează toate cuvintele de câte 3 litere din mulțimea  $\{a, b, c\}$ . Dacă primele patru cuvinte generate sunt, în această ordine: **aaa**, **aab**, **aac**, **aba**, care este cel de-al optulea cuvânt generat? **(4p.)**
- a. **acb**                      b. **acc**                      c. **aca**                      d. **bca**

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Subprogramul **f** are definiția alăturată. Ce valoare are **f(4)**? Dar **f(11)**? **(6p.)**
- ```
int f(int x)
{
    if(x<1) return 1;
    else return f(x-3)+1;
}
```
3. Scrieți programul **C/C++** care citește de la tastatură numărul natural **n** ( $0 < n < 100$ ) și un șir format din **n** numere reale. Aceste numere au atât partea întreagă cât și partea fracționară formate din cel mult trei cifre. Programul determină și afișează pe ecran toate numerele din șir care apar o singură dată în acesta.
- Exemplu:** dacă **n=7**, iar șirul este format din elementele (**3.4**, **-151**, **0.291**, **3.4**, **4.09**, **3.4**, **0.291**), atunci pe ecran se va afișa **-151 4.09**. **(10p.)**
4. Fișierul text **numere.txt** conține pe prima linie un număr natural **n** ( $0 < n < 100000$ ), iar pe a doua linie **n** numere naturale, formate din cel mult două cifre, separate prin câte un spațiu.
- a)** Scrieți un program **C/C++**, eficient din punct de vedere al timpului de executare, care afișează pe ecran toate numerele situate pe a doua linie a fișierului, în ordinea crescătoare a valorilor lor, separate prin câte un spațiu.
- Exemplu:** dacă fișierul **numere.txt** are următorul conținut:
- ```
7
12 21 22 11 9 12 3
```
- atunci pe ecran se va afișa: **3 9 11 12 12 21 22** **(6p.)**
- b)** Descrieți succint, în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența ei (3 – 4 rânduri). **(4p.)**

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Un program generează, în ordine crescătoare, numerele naturale de exact 5 cifre din mulțimea {1, 2, 3, 4, 5}. Fiecare dintre numerele generate are cifrele distincte două câte două. Primele 3 numere astfel generate sunt: 12345, 12354, 12435. Care este numărul generat imediat după 12543? (4p.)
- a. 15342                      b. 12534                      c. 13245                      d. 13452

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Funcția  $f$  are definiția alăturată. Dacă  $f(x)$  are valoarea 10100, care este valoarea lui  $x$ ? (6p.)
- ```
long f(int n)
{
    if(n<=0) return 0;
    else return f(n-1)+2*n;
}
```
3. Fișierul `bac.txt` conține pe prima linie un număr natural  $n$  ( $n \leq 100$ ), iar pe a doua linie, separate prin câte un spațiu,  $n$  numere naturale nenule, cu cel mult 4 cifre fiecare. Scrieți programul C/C++ care citește de la tastatură un număr natural  $k$  ( $k \leq 25$ ), construiește în memorie și afișează pe ecran un tablou unidimensional ce conține, în ordinea în care au fost citite, numerele de pe a doua linie a fișierului `bac.txt` care au cel puțin  $k$  divizori.
- Exemplu:** pentru  $k=5$  și fișierul `bac.txt` cu conținutul
- ```
6
100 9 400 56 7 10
```
- alăturat, tabloul care se afișează este: (10p.)
- 100 400 56
4. a) Scrieți în limbajul C/C++ doar antetul subprogramului `cifre`, care prin intermediul parametrului `nr` primește un număr natural de cel mult 9 cifre și furnizează prin intermediul parametrilor `nc` și `sc` numărul de cifre și respectiv suma cifrelor din scrierea lui `nr`. (4p.)
- b) Scrieți programul C/C++ care citește de la tastatură un număr natural  $n$  ( $10 \leq n \leq 10^9$ ) și verifică, folosind apeluri utile ale subprogramului `cifre`, dacă în scrierea în baza 10 a lui  $n$  se găsește cel puțin o cifră care să fie media aritmetică a celorlalte cifre din componența lui  $n$ . Programul afișează pe ecran mesajul **DA** în caz afirmativ și mesajul **NU** în caz contrar.
- Exemplu:** pentru  $n=27989$  programul va afișa mesajul **DA**, deoarece în scrierea lui  $n$  apare cifra 7 care este media aritmetică a celorlalte cifre din scrierea lui  $n$ :  $7 = (2+9+8+9)/4$ . Pentru  $n=7351$  se va afișa mesajul **NU**. (6p.)

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Se consideră opt bancnote: trei cu valoarea de 1 leu, două cu valoarea de 10 lei și trei cu valoarea de 100 de lei. Câte rezultate distincte se pot obține însumând valorile a exact cinci dintre cele opt bancnote, astfel încât suma să fie de cel puțin 200 de lei? **(4p.)**
- a. 6                      b. 12                      c. 15                      d. 3

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Se consideră subprogramul `f` definit alăturat. Ce valoare are `f(5)`? Dar `f(100)`? **(6p.)**
- ```
long f(int n)
{
    if(n<0) return 0;
    else    return f(n-2)+n;
}
```
3. Fișierul text `bac.txt` conține pe prima linie un număr natural `n` ( $n < 100$ ), iar pe a doua linie, separate prin câte un spațiu, `n` numere naturale de cel mult 8 cifre fiecare. Scrieți un program C/C++ care citește de la tastatură un număr natural `k` ( $k < 50$ ) precum și numerele din fișierul `bac.txt` și afișează pe ecran, cu câte un spațiu între ele, toate numerele de pe a doua linie a fișierului care sunt puteri ale lui `k`. Un număr natural `x` este putere a lui `k` dacă există un număr natural `y` astfel încât  $x = k^y$ . Dacă nu există un asemenea număr pe a doua linie a fișierului, se afișează pe ecran mesajul `NU`.  
**Exemplu:** dacă se citește de la tastatură `k=2`, iar fișierul are conținutul alăturat, atunci pe ecran se afișează numerele:
- |    |     |    |     |     |     |   |   |    |
|----|-----|----|-----|-----|-----|---|---|----|
| 8  | 32  | 56 | 317 | 809 | 256 | 2 | 1 | 60 |
| 32 | 256 | 2  | 1   |     |     |   |   |    |
- (10p.)**
4. a) Scrieți în limbajul C/C++ doar antetul unui subprogram `cif`, care primește prin intermediul primului parametru, `nr`, un număr natural cu cel mult 9 cifre și furnizează prin intermediul celui de-al doilea parametru, `s`, suma cifrelor din scrierea lui `nr`. **(4p.)**
- b) Scrieți programul C/C++ care citește de la tastatură un număr natural `n` ( $0 < n < 25$ ), apoi un șir de `n` numere naturale nenule cu cel mult 9 cifre fiecare și care afișează pe ecran, separate prin câte un spațiu, numerele din șir care au suma cifrelor maximă, folosind apeluri utile ale subprogramului `cif`.  
**Exemplu:** dacă pentru `n=8` se citește șirul de numere 274 56018 354 8219 293 287 932 634 atunci, pe ecran, se afișează numerele 56018 8219. **(6p.)**

**Subiectul III (30 de puncte)**

Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

- |                                                                                                                |                                                                                |
|----------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|
| <p>1. Se consideră subprogramul <code>f</code> definit alăturat. Ce valoare are <code>f(250)</code>? (4p.)</p> | <pre>int f(int x) {     if(x%3==0) return 0;     else return 1+f(x/3); }</pre> |
| a. 5                                                                                                           | b. 2                                                                           |
| c. 3                                                                                                           | d. 4                                                                           |

Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.

2. Se generează prin metoda backtracking mulțimile distincte ale căror elemente sunt numere naturale nenule și care au proprietatea că suma elementelor fiecărei mulțimi este egală cu 7. Astfel, sunt generate, în această ordine, mulțimile:  $\{1, 2, 4\}$ ,  $\{1, 6\}$ ,  $\{2, 5\}$ ,  $\{3, 4\}$ ,  $\{7\}$ . Folosind aceeași metodă pentru a genera mulțimile distincte ale căror elemente sunt numere naturale nenule și care au proprietatea că suma elementelor fiecărei mulțimi este egală cu 9, stabiliți în ce ordine sunt generate următoarele mulțimi:  $M1=\{2, 3, 4\}$ ;  $M2=\{3, 6\}$ ;  $M3=\{2, 7\}$ ;  $M4=\{4, 5\}$ . (6p.)
3. Se consideră subprogramul `cmdiv` care primește prin parametrii `x` și `y` două valori întregi pozitive ( $0 < x < 100$  și  $0 < y < 100$ ) și returnează cel mai mare divizor comun al lor.
- a) Scrieți definiția completă a subprogramului `cmdiv`. (4p.)
- b) Scrieți un program C/C++ care citește de la tastatură un număr natural `n` ( $1 \leq n \leq 100$ ) și determină, utilizând apeluri ale subprogramului `cmdiv`, numărul fracțiilor ireductibile de forma  $\frac{x}{n}$  cu `x` număr natural ( $1 \leq x \leq n$ ). Numărul obținut va fi afișat pe ecran.
- Exemplu:** pentru `n=6` se afișează 2, deoarece, dintre fracțiile  $\frac{1}{6}, \frac{2}{6}, \frac{3}{6}, \frac{4}{6}, \frac{5}{6}, \frac{6}{6}$  doar două sunt ireductibile :  $\frac{1}{6}$  și  $\frac{5}{6}$ . (6p.)
4. Fișierul text `bac.in` conține pe prima linie un număr natural `n` ( $0 < n < 5000$ ), iar pe a doua linie, separate prin câte un spațiu, `n` numere naturale, formate din cel mult 4 cifre fiecare. Scrieți un program C/C++ care determină și scrie în fișierul `bac.out` toate numerele de pe a doua linie a fișierului care apar o singură dată pe această linie. Numerele determinate se vor afișa în ordinea crescătoare a valorilor lor, separate prin câte un spațiu.
- Exemplu:** dacă pe prima linie a fișierului `bac.in` se află 10, iar pe linia a doua se găsesc numerele 2 4548 568 4548 57 89 5974 2 89 32 atunci valorile căutate sunt 32 57 568 5974. (10p.)



**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Se generează în ordine strict crescătoare numerele de câte șase cifre care conțin: cifra **1** o singură dată, cifra **2** de două ori și cifra **3** de trei ori. Se obțin, în această ordine, numerele: 122333, 123233, 123323, ..., 333221. Câte numere generate prin această metodă au prima cifră **1** și ultima cifră **2**? **(4p.)**
- a. 1                                      b. 3                                      c. 4                                      d. 8

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Pentru o valoare naturală strict mai mare decât 1 memorată în variabila globală **n**, subprogramul recursiv alăturat afișează cel mai mare divizor al lui **n**, mai mic decât **n**, la apelul **divi(n)**. Cu ce trebuie înlocuite **α** și **β**? **(6p.)**
- ```
void divi(long i)
{
    if(α==0)
        cout<< β; | printf("%ld", β);
    else divi(i-1);
}
```
3. Scrieți un program C/C++ care citește de la tastatură un număr natural **n** ( $1 \leq n \leq 50$ ) și apoi un șir de **n** numere naturale cu cel mult 4 cifre fiecare și care verifică dacă elementele șirului pot fi rearanjate astfel încât să respecte regula: al doilea element este cu 1 mai mare decât primul, al treilea cu 2 mai mare decât al doilea, ... , ultimul este cu **n-1** mai mare decât penultimul. Programul afișează pe ecran mesajul **DA** în caz afirmativ și mesajul **NU** în caz contrar.  
**Exemplu:** pentru **n=4** și șirul 8,5,11,6 se afișează **DA** (elementele pot fi rearanjate astfel încât să respecte regula dată: 5, 6, 8, 11) **(10p.)**
4. Se consideră subprogramul **cmax** care prin parametrul **a** primește un număr natural nenul mai mic decât 30000, iar prin parametrul **b** furnizează cifra maximă din numărul **a**.
- a) Scrieți, folosind limbajul C/C++, doar antetul subprogramului **cmax**. **(4p.)**
- b) Fișierul **bac.txt** conține cel mult 1000 numere naturale nenule, mai mici decât 30000 fiecare, separate prin câte un spațiu. Scrieți programul C/C++ care citește din fișierul **bac.txt** toate numerele și care determină cea mai mare cifră din scrierea lor și cel mai mic dintre numerele care conțin această cifră, folosind apeluri utile ale subprogramului **cmax**. Cifra și numărul determinate se vor afișa pe ecran, separate printr-un spațiu.  
**Exemplu:** dacă fișierul **bac.txt** conține valorile: 23 12 64 12 72 345 67 23 71 634 atunci pe ecran se afișează 7 67. **(6p.)**

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

- |  |  |
|--|--|
| <p>1. Valorile memorate de componentele tabloului <code>v</code>, cu indicii de la 0 la 5, sunt, în această ordine: 973, 51, 75, 350, 350, 15. Se consideră subprogramul <code>t</code> cu definiția alăturată. Care dintre următoarele expresii are valoarea 2 ? <b>(4p.)</b></p> <p>a. <code>t(0,v)+t(3,v)</code></p> <p>c. <code>t(4,v)+t(5,v)</code></p> | <pre>int t (int i, int v[]) {     if(i==0) return 0;     if(v[i]!=v[i-1]) return t(i-1,v);     return 1; }</pre> <p>b. <code>t(1,v)+t(4,v)</code></p> <p>d. <code>t(3,v)+t(4,v)</code></p> |
|--|--|

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Se generează în ordine strict crescătoare toate numerele de câte șase cifre care conțin: cifra 1 o singură dată, cifra 2 de două ori și cifra 3 de trei ori. Se obțin, în această ordine, numerele: 122333, 123233, 123323, ..., 333221. Ce număr se află imediat înaintea și ce număr se află imediat după numărul 332312 în șirul numerelor generate? **(6p.)**
3. Se consideră subprogramul `divxy` care primește prin parametrii `x` și `y` două valori întregi pozitive ( $0 < x < 1000$  și  $0 < y < 1000$ ) și returnează valoarea 1 dacă `y` este divizor al lui `x` sau `x` este divizor al lui `y` și returnează valoarea 0 în caz contrar.
- a) Scrieți definiția completă a subprogramului `divxy`. **(4p.)**
- b) Scrieți un program C/C++ care citește de la tastatură trei numere naturale nenule `a`, `b` și `n`, cu cel mult 3 cifre fiecare și care afișează pe ecran toți divizorii lui `n` din intervalul închis determinat de `a` și `b` folosind apeluri utile ale subprogramului `divxy`. Intervalul închis determinat de `a` și `b` este `[a,b]` dacă `a < b` sau `[b,a]` dacă `b ≤ a`. Numerele afișate sunt separate prin câte un spațiu. Dacă nu există niciun astfel de număr se afișează mesajul **NU EXISTA**.
- Exemplu:** pentru `a=85`, `b=10` și `n=40` se afișează: 10 20 40 (nu neapărat în această ordine). **(6p.)**
4. Fișierul `bac.in` conține pe prima linie un număr natural `n` ( $0 < n < 5000$ ), iar pe a doua linie, separate prin câte un spațiu, `n` numere naturale, formate din cel mult 4 cifre fiecare. Scrieți un program C/C++ care determină și scrie în fișierul `bac.out`, toate numerele, citite de pe a doua linie a fișierului `bac.in`, care apar de cel puțin două ori. Numerele determinate se vor scrie în ordine crescătoare, pe aceeași linie, separate prin câte un spațiu.
- Exemplu:** dacă fișierul `bac.in` conține pe prima linie numărul 11, iar pe linia a doua valorile 23 12 54 12 78 345 67 23 78 934 23  
atunci fișierul `bac.out` va conține: 12 23 78 **(10p.)**

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Care este antetul corect al subprogramului `cifre`, care primește prin intermediul primului parametru, `x`, un număr natural și furnizează prin intermediul celui de-al doilea parametru, `y`, suma cifrelor numărului natural `x`? **(4p.)**
- a. `void cifre(int x, int &y)`                      b. `int cifre(int x)`  
c. `void cifre(int x, int y)`                      d. `void cifre(int &x, int y)`

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Se consideră numărul `x=21034085`. Permutând cifrele lui `x` se obțin alte numere naturale.
- a) Dați exemplu de un astfel de număr care să fie divizibil cu 25.  
b) Câte dintre numerele obținute au exact 7 cifre? **(6p.)**

3. Se consideră subprogramul `dist2`, care primește prin intermediul parametrilor `xa`, `ya` și respectiv `xb`, `yb`, coordonatele carteziene întregi (abscisă, ordonată) pentru două puncte din plan, `A` și respectiv `B`. Subprogramul returnează pătratul distanței dintre cele două puncte.

a) Scrieți în limbajul C/C++ definiția completă a subprogramului `dist2`. **(4p.)**

b) Scrieți un program C/C++ care citește de la tastatură 8 valori întregi reprezentând coordonatele carteziene pentru patru puncte din plan și afișează mesajul `Da` dacă cele patru puncte pot fi vârfurile unui pătrat, iar în caz contrar afișează mesajul `Nu`, folosind apeluri utile ale subprogramului `dist2`.

**Exemplu:** dacă coordonatele punctelor sunt cele alăturate  
atunci se va afișa mesajul `Da` **(6p.)**

0	0
3	0
3	3
0	3

4. Fișierul text `date.in` conține pe prima linie un număr natural nenul `n` ( $n \leq 100$ ), iar pe a doua linie `n` numere naturale nenule, separate prin câte un spațiu, fiecare număr având maximum 4 cifre. Scrieți un program C/C++ care citește toate numerele din fișierul text `date.in`, construiește în memorie un tablou unidimensional `a`, cu cele `n` elemente din fișier și afișează pe ecran perechile  $(a_i, a_j)$ ,  $1 \leq i < j \leq n$ , cu proprietatea că elementele fiecăreia dintre aceste perechi au aceeași paritate. Fiecare pereche se va afișa pe câte o linie a ecranului, elementele perechii fiind separate prin câte un spațiu. În cazul în care nu există nicio pereche, se va afișa valoarea 0.

**Exemplu:** dacă fișierul `date.in` are conținutul alăturat,  
se vor afișa:

16 22

16 6

22 6

3 1

**(10p.)**

5	
16	22 3 6 1

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Utilizând metoda backtracking, se generează în ordine lexicografică toate anagramele cuvântului **caiet** (cuvinte formate din aceleași litere, eventual în altă ordine). Câte cuvinte care încep cu litera **t** vor fi generate? (4p.)
- a. 1                                      b. 6                                      c. 12                                      d. 24

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Se consideră subprogramul **f**, definit alături. Ce valoare are **f(4)**? Dar **f(9)**? (6p.)
- ```
int f(int n)
{
    if (n==0) return 0;
    if(n%2==1) return n-f(n-1);
    return f(n-1)-n;
}
```

```
int f(int n)
{
    if (n==0) return 0;
    if(n%2==1) return n-f(n-1);
    return f(n-1)-n;
}
```
3. Subprogramul **mult**, cu doi parametri, primește prin intermediul primului parametru, **n**, un număr natural nenul cu maximum trei cifre și prin intermediul celui de-al doilea parametru, **a**, un tablou unidimensional având **n** componente numere naturale cu cel mult 8 cifre fiecare. Subprogramul returnează valoarea 1 dacă cele **n** componente ale lui **a** pot forma o mulțime și returnează 0 în caz contrar.
- a) Scrieți definiția completă a subprogramului **mult**. (6p.)
- b) Fișierul text **date.in** conține cel mult 400 de numere naturale având maximum 8 cifre fiecare. Scrieți un program C/C++ care, folosind apeluri utile ale subprogramului **mult**, afișează pe ecran valoarea maximă **k**, astfel încât primele **k** numere succesive din fișier să poată forma o mulțime.
- Exemplu:** dacă fișierul **date.in** conține
- 16 17 8 31 8 2 10
- atunci se va afișa 4 (deoarece primele patru numere din fișier pot forma o mulțime și acesta este cardinalul maxim posibil în condițiile impuse de enunțul problemei) (4p.)
4. Scrieți un program C/C++ care citește de la tastatură două numere naturale nenule **n** și **k** ( $k < n \leq 10000$ ) și afișează pe ecran un șir format din **k** numere naturale consecutive care au suma **n**. Numerele se vor afișa pe ecran, în ordine crescătoare, despărțite între ele prin câte un spațiu. În cazul în care nu există un astfel de șir, se va afișa mesajul **Nu există**. (10p.)





**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Utilizând metoda backtracking se generează toate matricele pătratice de ordinul 4 ale căror elemente aparțin mulțimii  $\{0,1\}$ , cu proprietatea că pe fiecare linie și pe fiecare coloană există o singură valoare 1. Primele 4 soluții generate sunt, în această ordine:

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

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

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

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

Care este a opta soluție?

(4p.)

a. 0 1 0 0  
1 0 0 0  
0 0 0 1  
0 0 1 0

b. 0 1 0 0  
1 0 0 0  
0 0 1 0  
0 0 0 1

c. 0 1 0 0  
0 0 1 0  
1 0 0 0  
0 0 0 1

d. 0 0 1 0  
1 0 0 0  
0 1 0 0  
0 0 0 1

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Se consideră subprogramul `f`, definit alăturat.

a) Ce valoare are `f(25)`?

b) Dar expresia `f(1)+f(5)+f(15)`? (6p.)

```
int f(int n)
{ if (n>20) return 0;
  else return 5+f(n+5);
}
```

3. Se consideră subprogramul `cifre`, care primește prin intermediul primului parametru, `a`, un număr natural cu maximum 8 cifre nenule și returnează, prin intermediul celui de-al doilea parametru `b`, cel mai mic număr care se poate forma cu toate cifrele distincte ale lui `a`.

a) Scrieți definiția completă a subprogramului `cifre`. (4p.)

b) Se consideră fișierul text `date.in` ce conține pe prima linie un număr natural nenul `n` ( $n \leq 100$ ), iar pe a doua linie `n` numere naturale, separate prin câte un spațiu, fiecare număr având maximum 8 cifre nenule. Scrieți un program C/C++ care citește toate numerele din fișierul text `date.in` și afișează pe ecran, despărțite prin câte un spațiu, numerele situate pe a doua linie a fișierului, formate numai din cifre distincte ordonate strict crescător, folosind apeluri utile ale subprogramului `cifre`. În cazul în care nu există niciun astfel de număr se va afișa valoarea 0.

**Exemplu:** dacă fișierul `date.in` are conținutul

|   |    |     |     |        |     |     |
|---|----|-----|-----|--------|-----|-----|
| 6 | 16 | 175 | 333 | 242477 | 321 | 269 |
|---|----|-----|-----|--------|-----|-----|

alăturat, atunci se vor afișa numerele: 16 269 (6p.)

4. Scrieți un program C/C++ care citește de la tastatură un număr natural nenul `n` ( $n \leq 100$ ) și `2*n` numere naturale de maximum 3 cifre; primele `n` reprezintă elementele tabloului unidimensional `a`, iar următoarele `n` elementele tabloului unidimensional `b`; fiecare tablou are elementele numerotate începând de la 1. Programul construiește în memorie și afișează pe ecran, cu spații între ele, cele `n` elemente ale unui tablou unidimensional `c`, în care orice element `ci` ( $1 \leq i \leq n$ ) se obține conform definiției următoare:

$c_i = \begin{cases} a_i \text{ concatenat cu } b_i, & \text{dacă } a_i < b_i \\ b_i \text{ concatenat cu } a_i, & \text{altfel} \end{cases}$

**Exemplu:** dacă se citesc `n=3`, `a=(12,123,345)` și `b=(1,234,15)`, atunci se afișează elementele tabloului `c` astfel:

112 123234 15345

(10p.)

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Pentru a genera toate numerele naturale cu exact 4 cifre și care au cifrele în ordine strict descrescătoare, se poate utiliza un algoritm echivalent cu cel pentru generarea: **(4p.)**
- a. aranjamentelor de 4 obiecte luate câte 10      b. combinărilor de 10 obiecte luate câte 4
- c. permutărilor a 10 obiecte                              d. permutărilor a 4 obiecte

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Se consideră subprogramul `f`, definit alăturat. Ce valoare are `f(12,3)`? Dar `f(21114,1)`? **(6p.)**
- ```
int f(int n,int c)
{
    if(n==0)return 0;
    if(n%10==c)
        return n%100+f(n/10,c);
    return f(n/10,c);
}
```
3. Fișierul text `numere.txt` conține, pe o singură linie, cel mult 1000 de numere naturale nenule cu cel mult 4 cifre fiecare, numerele fiind separate prin câte un spațiu. Scrieți un program C/C++ care citește toate numerele din fișier și scrie pe ecran toate numerele pare citite, ordonate crescător. Dacă fișierul `numere.txt` nu conține niciun număr par, atunci se va afișa pe ecran mesajul `nu exista`.  
**Exemplu:** dacă fișierul `numere.txt` conține numerele 2 3 1 4 7 2 5 8 6, atunci pe ecran se va afișa: 2 2 4 6 8 **(10p.)**
4. Se consideră subprogramele
- `prim`, care primește prin intermediul unicului său parametrului `x` un număr natural nenul de cel mult 4 cifre și returnează valoarea 1 dacă `x` este un număr prim și 0 în caz contrar;
  - `numar`, care primește prin intermediul parametrului `x` un număr natural nenul de cel mult 4 cifre și furnizează prin intermediul parametrului `nrp` numărul de numere prime mai mici decât `x`.
- a) Scrieți numai antetul subprogramului `prim` și definiția completă a subprogramului `numar`. **(6p.)**
- b) Scrieți un programul C/C++ în care se citesc de la tastatură două numere naturale nenule de cel mult 4 cifre, `a` și `b` (`a < b`), și, prin apeluri utile ale subprogramului `numar`, se verifică dacă intervalul închis `[a,b]` conține cel puțin un număr prim. Programul va afișa pe ecran, în caz afirmativ, mesajul `DA`, iar în caz contrar, mesajul `NU`. **(4p.)**



**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Se utilizează metoda backtracking pentru a genera în ordine lexicografică toate cuvintele de câte patru litere din mulțimea {**d, a, n, s**}, astfel încât în niciun cuvânt să nu existe două litere alăturate identice. Știind că primele trei cuvinte generate sunt, în ordine, **ada**d, **ada**n și **ada**s, care va fi ultimul cuvânt obținut? **(4p.)**
- a. **snns**                      b. **nsns**                      c. **snsn**                      d. **dans**

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Se consideră subprogramul **f**, definit alăturat. Ce se va afișa la apelul **f(38);**? **(6p.)**

```
void f(int x)
{
    if(x) {
        if(x%3==0){
            cout<<3; | printf("3");
            f(x/3);
        }
        else{
            f(x/3);
            cout<<x%3; | printf("%d",x%3);
        }
    }
}
```

3. Fișierul text **INTRARE.TXT** conține, pe o singură linie, cel mult 100 de numere naturale nenule de cel mult patru cifre fiecare, numerele fiind separate prin câte un spațiu. Scrieți un program C/C++ care citește numerele din fișier și scrie în fișierul text **IESIRE.TXT**, în ordine crescătoare, toate valorile distincte obținute ca sumă de două elemente distincte aflate pe prima linie a fișierului **INTRARE.TXT**.

**Exemplu:**

**INTRARE.TXT**

1 4 3 2

**IESIRE.TXT**

3 4 5 6 7

**(10p.)**

4. Se consideră subprogramul **multiplu**, cu doi parametri, care:
- primește prin intermediul parametrilor **a** și **k** două numere întregi de cel mult 4 cifre;
  - returnează cel mai mic multiplu al lui **k** mai mare sau egal cu **a**.

**a) Scrieți numai antetul subprogramului **multiplu**.**

**(4p.)**

**b) Scrieți declarațiile de date și programul principal C/C++ care citește de la tastatură trei numere naturale nenule **x, y, z**, de cel mult 4 cifre fiecare, (**x ≤ y**), și care, prin apeluri utile ale subprogramului **multiplu**, verifică dacă intervalul [**x, y**] conține cel puțin un multiplu al lui **z**. Programul va afișa pe ecran, în caz afirmativ, mesajul **DA**, iar în caz contrar mesajul **NU**.**

**(6p.)**

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Se utilizează metoda backtracking pentru a genera în ordine lexicografică toate cuvintele de câte trei litere distincte din mulțimea {d,a,n,s}. Care este cel de-al treilea cuvânt obținut? (4p.)

a. ads                      b. ans                      c. dan                      d. and

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Se consideră subprogramul `f`, definit alăturat. Ce valoare are `f(88,1)`? Dar `f(3713,3)`? (6p.)
- ```
int f(int n,int c){
    if(n==0)
        return 0;
    if(n%10==c)
        return f(n/10,c)*10+c;
    return f(n/10,c);
}
```
3. Fișierul text **BAC.TXT** conține, pe o singură linie, cel puțin 3 și cel mult 100 de numere naturale nenule distincte de cel mult 4 cifre fiecare, numerele fiind separate prin câte un spațiu. Scrieți un program C/C++ care citește toate numerele din fișierul **BAC.TXT** și scrie pe ecran, în ordine descrescătoare, cele mai mici 3 numere citite.  
**Exemplu:** dacă fișierul **BAC.TXT** conține numerele 1017 48 310 5710 162, atunci se va afișa: 310 162 48 (10p.)
4. Se consideră subprogramul `divizor`, care:
- primește prin intermediul parametrului `a` un număr natural strict mai mare decât 1, de cel mult 4 cifre;
  - furnizează prin intermediul parametrului `d` cel mai mare divizor al lui `a` strict mai mic decât `a`.
- a) Scrieți numai antetul subprogramului `divizor`. (4p.)
- b) Scrieți declarațiile de date și programul principal C/C++ care citește de la tastatură un număr natural nenul `x`, de cel mult 4 cifre și, prin apeluri utile ale subprogramului `divizor`, verifică dacă `x` este număr prim. Programul va afișa pe ecran în caz afirmativ mesajul **DA**, iar în caz contrar mesajul **NU**. (6p.)

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Se utilizează metoda backtracking pentru a genera în **ordine lexicografică** toate cuvintele care conțin toate literele din mulțimea  $\{a, m, i, c\}$ , astfel încât fiecare literă să apară exact o dată într-un cuvânt. Câte soluții sunt generate după cuvântul **amic** și înainte de cuvântul **cam**?
- a. 6                                      b. 4                                      c. 1                                      d. 3                                      **(4p.)**

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Se consideră subprogramul **f**, definit alături. Ce valoare are **f(12,5)**? Dar **f(261,31)**? **(6p.)**
- ```
int f(int a,int b)
{
    if(a<10)
        return b;
    return f(a/10,b)*10+b+1;
}
```
3. Fișierul text **bac.txt** conține, pe o singură linie, cel puțin 3 și cel mult 100 de numere naturale nenule distincte de cel mult 4 cifre fiecare, numerele fiind separate prin câte un spațiu. Scrieți un program C/C++ care citește numerele din fișier și scrie pe ecran ultima cifră a produsului celor mai mari 3 numere citite.  
**Exemplu:** dacă fișierul **bac.txt** conține numerele:  
1017 48 312 5742 162  
atunci se va afișa: 8 (ultima cifră a produsului numerelor 1017, 5742, 312) **(10p.)**
4. Se consideră subprogramul **divizor**, care:
- primește prin intermediul parametrului **a**, un număr natural nenul de cel mult 4 cifre, strict mai mare ca 1;
  - furnizează prin intermediul parametrului **d**, cel mai mic divizor al lui **a** strict mai mare decât 1.
- a) Scrieți numai antetul subprogramului **divizor**. **(4p.)**
- b) Scrieți declarațiile de date și programul principal C/C++ care citește de la tastatură un număr natural **x** (**x>1**), și, prin apeluri utile ale subprogramului **divizor**, verifică dacă **x** este număr prim. Programul va afișa pe ecran, în caz afirmativ, mesajul **DA**, iar în caz contrar mesajul **NU**. **(6p.)**

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Se utilizează metoda backtracking pentru a genera toate cuvintele care conțin toate literele din mulțimea  $\{i, n, f, o\}$ , astfel încât fiecare literă să apară exact o dată într-un cuvânt și literele  $n$  și  $o$  să nu se afle pe poziții vecine. Știind că primul cuvânt generat este **info**, iar al treilea, al patrulea și al cincilea sunt **nifo**, **niof**, **nfio** care este cel de-al doilea cuvânt obținut? **(4p.)**
- a. **iofn**                      b. **inof**                      c. **ionf**                      d. **niof**

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Se consideră subprogramul **f**, definit alăturat. Ce valoare are **f(3,13)**? Dar **f(1000,2009)**? **(6p.)**
- ```
int f(int a,int b)
{
    if(2*a>=b)
        return 0;
    if(b%a==0)
        return b-a;
    return f(a+1,b-1);
}
```
3. Fișierul text **bac.txt** conține, pe prima sa linie, 100 de numere naturale de cel mult 4 cifre fiecare, numerele fiind ordonate crescător și separate prin câte un spațiu, iar pe a doua linie un singur număr natural **x**, cu cel mult 4 cifre. Scrieți un program **C/C++** care citește toate numerele din fișier și verifică dacă **x** se află în șirul celor 100 de numere aflate pe prima linie a fișierului. În caz afirmativ, se va afișa pe ecran mesajul **DA**, altfel se va afișa mesajul **NU**.
- Exemple:** dacă fișierul **bac.txt** conține:
- ```
17 38 40 45 50 51 52 53 54 55 ... 145
52
```
- atunci se va afișa: **DA** ;
- dacă fișierul **bac.txt** conține:
- ```
2 11 15 16 20 25 30 35 40 ... 495
33
```
- atunci se va afișa: **NU**. **(10p.)**
4. Se consideră subprogramul **radical1**, care:
- primește prin intermediul parametrului **a**, un număr natural nenul de cel mult 4 cifre;
  - furnizează prin intermediul parametrului **x** cel mai mare număr natural cu proprietatea că  $x^2$  este mai mic sau egal cu **a**; de exemplu, dacă **a=20**, subprogramul va furniza prin **x** valoarea 4.
- a) Scrieți numai antetul subprogramului **radical1**. **(4p.)**
- b) Scrieți declarațiile de date și programul principal **C/C++** care citește de la tastatură un număr natural nenul de cel mult 4 cifre, **n**, și prin apeluri utile ale subprogramului **radical1**, verifică dacă **n** este pătrat perfect. Programul va afișa pe ecran în caz afirmativ mesajul **DA**, iar în caz contrar mesajul **NU**. **(6p.)**

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Generarea matricelor pătrate de ordinul  $n$ , cu elemente 0 și 1, cu proprietatea că pe fiecare linie și pe fiecare coloană există un singur element egal cu 1, se poate realiza utilizând metoda backtracking. Algoritmul utilizat este echivalent cu algoritmul de generare a:
- a. combinărilor                      b. permutărilor                      c. aranjamentelor                      d. produsului cartezian
- (4p.)**

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Se consideră tabloul unidimensional  $a$  definit global, ce memorează elementele  $a_1=12$ ,  $a_2=35$ ,  $a_3=2$ ,  $a_4=8$  și subprogramul  $f$ , definit alăturat. Ce valoare are  $f(1)$ ? Dar  $f(4)$ ? **(6p.)**
- ```
int f(int x)
{
    if(x>=1)
        return (a[x]+f(x-1))%10;
    else
        return 0;
}
```
3. Subprogramul `cifra` primește prin parametrul  $x$  un număr real nenul pozitiv și furnizează prin parametrul  $y$  valoarea cifrei unităților părții întregi a lui  $x$ .  
**Exemplu:** dacă  $x=34.567$ , după apel  $y=4$ .
- a) Scrieți definiția completă a subprogramului `cifra`. **(6p.)**
- b) Scrieți un program C/C++ care citește de la tastatură două numere reale cu cel mult două zecimale, numere reprezentând mediile semestriale obținute de un elev. Programul stabilește, folosind apeluri utile ale subprogramului `cifra`, dacă cele două medii citite se află în aceeași categorie de medii sau nu. Precizăm că orice medie, în funcție de intervalul în care se încadrează, face parte din una dintre categoriile:  $[3,3.99]$ ,  $[4,4.99]$ ,  $[5,5.99]$ ,  $[6,6.99]$ ,  $[7,7.99]$ ,  $[8,8.99]$  sau  $[9,10]$ . În cazul în care ambele medii fac parte din aceeași categorie, programul va afișa mesajul **Da**, altfel va afișa mesajul **Nu**. **(4p.)**
4. În fișierul `text.in` se află un text scris pe mai multe linii, pe fiecare linie fiind caractere diverse: litere mici ale alfabetului englez, cifre, spații și semne de punctuație. Găsiți o metodă eficientă din punctul de vedere al timpului de executare și al gestionării memoriei ce permite afișarea celor mai frecvente perechi de vocale alăturate din text. Menționăm că vocalele sunt:  $a, e, i, o$  și  $u$ .  
De exemplu, dacă `text.in` conține textul:  
Aleea ce strabate valea  
e-o unduire de pietris scanteietor,  
De-aceea nu stii daca zboara sau doar inoata in unde aurii  
cele mai frecvente perechi de vocale întâlnite în acest text sunt: ea și oa (ele apar de 3 ori, spre deosebire de alte perechi de vocale alăturate care apar de mai puține ori).
- a) Descrieți succint, în limbaj natural, metoda de rezolvare aleasă, explicând în ce constă eficiența ei (3 – 4 rânduri). **(4p.)**
- b) Scrieți programul C/C++ corespunzător metodei descrise la cerința a. **(6p.)**

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Utilizând metoda backtracking pentru afișarea tuturor modalităților de descompunere a unui număr natural ca o sumă de numere naturale nenule, pentru  $n=3$  se obțin, în ordine, soluțiile:  $1+1+1$ ;  $1+2$ ;  $2+1$ ; 3. Ordinea de scriere a termenilor dintr-o descompunere este semnificativă. Folosind aceeași metodă pentru  $n=10$ , care este soluția generată imediat după  $1+1+3+5$ ? **(4p.)**

a.  $1+1+4+1+1+1+1$       b.  $1+1+7+1$       c.  $1+2+7$       d.  $1+1+4+4$

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Considerăm subprogramul `f`, definit alăturat. Care va fi valoarea variabilei globale `x` după apelul `f(4962,x)`, dacă înainte de apel, `x` are valoarea 0? Dar dacă înainte de apel `x` are valoarea 52? **(6p.)**
- ```
void f(int n,int &a)
{int c;
 if(n!=0){
  c=n%10;
  if(a<c) a=c;
  f(n/10,a);
 }
```
3. Scrieți definiția completă a unui subprogram `fib` cu doi parametri, `n` și `v`, care primește prin intermediul parametrului `n` un număr natural ( $1 < n < 30$ ) și returnează prin intermediul parametrului `v` un tablou unidimensional care conține primii `n` termeni **impari** ai șirului lui Fibonacci (amintim că șirul lui Fibonacci este:  $1, 1, 2, 3, 5, 8, 13, 21, \dots$ ). **(10p.)**
4. a) Fișierul `date.in` conține un șir de cel mult 10000 numere naturale (printre care cel puțin un număr par și cel puțin un număr impar), cu cel mult 2 cifre fiecare, separate prin câte un spațiu. Scrieți un program C/C++ care citește numerele din fișierul `date.in` și scrie în fișierul text `date.out` valorile distincte citite, separate prin câte un spațiu, respectându-se regula: pe prima linie vor fi scrise numerele impare în ordine crescătoare, iar pe linia a doua numerele pare, în ordine descrescătoare. Alegeți o metodă eficientă din punctul de vedere al timpului de executare. **(6p.)**
- Exemplu:** dacă pe prima linie a fișierului `date.in` se află numerele:  
75 12 3 3 18 75 1 3  
atunci fișierul `date.out` va conține:  
1 3 75  
18 12
- b) Descrieți succint, în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența ei (3 – 4 rânduri). **(4p.)**

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Se generează, prin metoda backtracking, toate partițiile mulțimii  $A = \{1, 2, 3\}$  obținându-se următoarele soluții:  $\{1\}\{2\}\{3\}$ ;  $\{1\}\{2, 3\}$ ;  $\{1, 3\}\{2\}$ ;  $\{1, 2\}\{3\}$ ;  $\{1, 2, 3\}$ . Se observă că dintre acestea, prima soluție e alcătuită din exact trei submulțimi. Dacă se folosește aceeași metodă pentru a genera partițiile mulțimii  $\{1, 2, 3, 4\}$  stabiliți câte dintre soluțiile generate vor fi alcătuite din exact trei submulțimi. **(4p.)**
- a. 3                      b. 12                      c. 6                      d. 5

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Se consideră tabloul unidimensional **a** definit global, ce memorează elementele  $a_1=1$ ,  $a_2=2$ ,  $a_3=0$  și subprogramul **f**, definit alăturat. Ce valoare are  $f(2,1)$ ? Dar  $f(3,3)$ ? **(6p.)**
- ```
int f(int b,int i)
{if(i>=1)
    return f(b,i-1)*b+a[i];
else return 0;
}
```
3. Subprogramul **verif** primește prin singurul său parametru, **x**, un număr natural nenul cu cel mult 9 cifre și returnează valoarea 1 dacă numărul conține cel puțin o secvență de 3 cifre impare alăturate și 0 în caz contrar.  
**Exemplu:** dacă  $x=7325972$  se va returna valoarea 1.  
**a)** Scrieți definiția completă a subprogramului **verif**. **(6p.)**  
**b)** Scrieți un program C/C++ care citește de la tastatură un număr natural nenul **n** cu exact 6 cifre și, folosind apeluri utile ale subprogramului **verif**, verifică dacă **n** are primele trei cifre impare. Programul afișează pe ecran mesajul **Da** în caz afirmativ și mesajul **Nu** în caz contrar.  
**Exemple:** dacă se citește  $n=132567$  se va afișa mesajul **Nu**, iar dacă  $n=979243$ , se va afișa mesajul **Da**. **(4p.)**
4. Pentru un șir de numere naturale, numim "nod" al șirului un termen din șir care are doi vecini, termenul precedent și termenul următor din șir, și valoarea termenului respectiv este strict mai mică decât suma valorilor celor doi vecini ai săi.  
**a)** Fișierul text **date.in** conține un șir de cel puțin două și cel mult 10000 de numere naturale având maximum 6 cifre fiecare, numere separate prin câte un spațiu. Scrieți un program C/C++ care citește toate numerele din fișier și afișează numărul de "noduri" ale șirului citit, folosind un algoritm eficient din punctul de vedere al memoriei utilizate. **(6p.)**  
**Exemplu:** dacă fișierul **date.in** are următorul conținut:  
51   20   100   43   43   618   5000   31   2020   114   116   4  
atunci pe ecran se afișează 7 (cele șapte numere subliniate reprezintă "noduri" ai șirului)  
**b)** Descrieți succint, în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența ei (3 – 4 rânduri). **(4p.)**

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Se generează, prin metoda backtracking, toate modalitățile de așezare a numerelor naturale de la 1 la 5, astfel încât oricare 2 numere consecutive să nu se afle pe poziții alăturate. Dacă primele două soluții sunt: (1,3,5,2,4) și (1,4,2,5,3), care este prima soluție generată în care primul număr este 4? **(4 p.)**
- a. (4, 1, 3, 2, 5)      b. (4, 2, 5, 1, 3)      c. (4, 3, 5, 3, 1)      d. (4, 1, 3, 5, 2)

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Se consideră subprogramul `f`, definit alăturat. Ce se va afișa în urma apelului `f(5,0);`? **(6p.)**
- ```
void f(int i,int j)
{ if(j<=9) f(i,j+1);
  printf("%d*%d=%d\n",i,j,i*j);/
  cout<<i<<'<'<j<<'='<<i*j<<endl;
}
```
3. Subprogramul `diviz` primește prin intermediul parametrului `n` un număr natural nenul ( $2 \leq n \leq 200$ ), iar prin intermediul parametrului `a`, un tablou unidimensional care conține `n` valori naturale nenule, fiecare dintre acestea având cel mult patru cifre. Elementele tabloului sunt numerotate de la 1 la `n`. Subprogramul returnează o valoare egală cu numărul de perechi  $(a_i, a_j)$ ,  $1 \leq i < j \leq n$ , în care  $a_i$  este divizor al lui  $a_j$ , sau  $a_j$  este divizor al lui  $a_i$ . Scrieți definiția completă a subprogramului `diviz`, în limbajul C/C++. **Exemplu:** pentru `n=5` și `a=(4,8,3,9,4)` subprogramul returnează valoarea 4. **(10p.)**
4. Fișierul text `date.in` conține pe prima linie, separate prin câte un spațiu, cel mult 1000 de numere naturale, fiecare dintre ele având maximum 9 cifre.
- a) Scrieți un program C/C++ care citește numerele din fișierul `date.txt` și determină cea mai lungă secvență ordonată strict descrescător, formată din valori citite consecutiv din fișier. Numerele din secvența găsită vor fi afișate pe ecran, pe o linie, separate prin câte un spațiu. Dacă sunt mai multe secvențe care respectă condiția impusă, se va afișa doar prima dintre acestea. Alegeți o metodă de rezolvare eficientă din punctul de vedere al timpului de executare.
- Exemplu:** dacă fișierul `date.in` conține
- |                        |                                    |
|------------------------|------------------------------------|
| 5 2 19 4 3 6 3 2 1 0 8 | pe ecran se afișează:<br>6 3 2 1 0 |
|------------------------|------------------------------------|
- (6p.)**
- b) Descrieți succint, în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența ei (3 – 4 rânduri). **(4p.)**



**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Se generează, prin metoda backtracking, toate modalitățile de așezare a numerelor naturale de la 1 la 5 astfel încât oricare două numere consecutive să nu se afle pe poziții alăturate. Dacă primele două soluții sunt: (1,3,5,2,4) și (1,4,2,5,3), care este prima soluție generată care începe cu 2? (4p.)
- a. (2, 4, 1, 3, 5)      b. (2, 5, 4, 3, 1)      c. (2, 4, 1, 3, 1)      d. (2, 3, 5, 4, 1)

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Se consideră funcția  $f$ , definită alăturat.  
Ce se afișează ca urmare a executării secvenței de mai jos în care variabilele  $a$  și  $b$  sunt de tip `int`?  
`a=4; b=18;`  
`printf("%d",f(a,b));/ cout<<f(a,b);`  
`printf("%d %d",a,b);/ cout<<a<<b;` (6p.)
- ```
int f( int &a, int &b)
{ while (a !=b)
  if (a>b) a=a-b;
  else b=b-a;
  return a;}
```
3. Subprogramul `sfx` primește prin singurul său parametru,  $x$ , un număr natural din intervalul  $[100, 2000000000]$  și returnează valoarea 1 dacă ultimele trei cifre ale numărului sunt în ordine strict descrescătoare sau valoarea 0 în caz contrar.  
**Exemplu:** dacă  $x=24973$  se va returna valoarea 1.
- a) Scrieți definiția completă a subprogramului `sfx`. (5p.)
- b) Scrieți un program C/C++ care citește de la tastatură un număr natural  $n$  format din exact 6 cifre și verifică, utilizând apeluri ale subprogramului `sfx`, dacă acest număr are toate cifrele în ordine strict descrescătoare. Programul va afișa mesajul **Da** în caz afirmativ și mesajul **Nu** în caz contrar.  
**Exemplu:** dacă  $n=756543$  se va afișa **Nu**, iar dacă  $n=976532$  se va afișa **Da**. (5p.)
4. Pentru un șir de numere naturale, numim "pol" al șirului un termen din șir care are doi vecini, termenul precedent și termenul următor din șir, și valoarea termenului respectiv este strict mai mare decât valoarea fiecăruia dintre cei doi vecini ai săi.
- a) Fișierul `text date.in` conține un șir de cel mult 10000 de numere naturale având maximum 6 cifre fiecare, numere separate prin câte un spațiu. Scrieți un program C/C++ care citește toate numerele din fișier și afișează numărul de "poli" ai șirului citit, folosind un algoritm eficient din punctul de vedere al memoriei utilizate. (6p.)
- Exemplu:** dacă fișierul `date.in` are următorul conținut:
- 51   20   100   43   43   618   5000   31   2020   114   116   4
- atunci pe ecran se afișează 4 (cele patru numere subliniate reprezintă "poli" ai șirului)
- b) Descrieți succint, în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența ei (3 – 4 rânduri). (4p.)





**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Se utilizează metoda backtracking pentru a genera toate submulțimile cu 4 elemente ale mulțimii  $\{1, 2, 3, 4, 5, 6\}$ . Numărul de submulțimi generate este: **(4p.)**
- a. 30                      b. 35                      c. 5                      d. 15

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Se consideră subprogramul recursiv definit alăturat. Ce valori vor fi afișate pe ecran în urma apelului `gama(6);`? **(6p.)**
- ```
void gama(int n)
{
    int i;
    if(n >= 3)
    {
        for(i = 3; i <= n; i++)
            printf("%d ", n); | cout << n << " ";
        gama(n - 3);
    }
}
```
3. a) Să se scrie definiția completă a subprogramului `calcul`, care primește prin intermediul celor doi parametri ai săi două numere întregi,  $n$  și  $k$  ( $1 \leq n \leq 1000000000$  și  $1 \leq k \leq 9$ ), și returnează cifra de rang  $k$  a numărului  $n$ . Rangul unei cifre este numărul său de ordine, numerotând cifrele de la dreapta la stânga; cifra unităților având rangul 1. Dacă numărul  $k$  este mai mare decât numărul de cifre ale lui  $n$ , atunci funcția returnează valoarea -1.  
**Exemplu:** dacă  $n=9243$  și  $k=3$ , în urma apelului se va returna 2. **(5p.)**
- b) Scrieți un program C/C++ care citește de la tastatură un număr natural  $n$  cu cel mult 8 cifre. Programul va verifica, utilizând apeluri ale subprogramului `calcul`, dacă orice cifră a lui  $n$  are rangul cifrei mai mare sau egal cu valoarea cifrei respective și va afișa mesajul **Da** în caz afirmativ și mesajul **Nu** în caz contrar.  
**Exemplu :** pentru  $n=4160$  se va afișa **Nu**. **(5p.)**
- |       |   |   |   |   |
|-------|---|---|---|---|
| rang  | 4 | 3 | 2 | 1 |
| cifră | 4 | 1 | 6 | 0 |
4. Fișierul text `SIR.TXT` conține pe prima linie un număr natural  $n$  ( $1 \leq n \leq 10000$ ) și pe a doua linie, separate prin spații, un șir **crescător** de  $n$  numere naturale cu cel mult 9 cifre fiecare.  
Numim platou într-un șir de valori o secvență de elemente identice situate pe poziții alăturate. Lungimea unui platou este egală cu numărul de elemente care îl formează.
- a) Scrieți un program C/C++ care citește valorile din fișier și, printr-o metodă eficientă din punct de vedere al timpului de executare și al spațiului de memorie utilizat afișează pe ecran, separate printr-un spațiu, lungimea maximă a unui platou, precum și valoarea care formează platoul. În cazul în care sunt mai multe platouri de aceeași lungime se va afișa valoarea cea mai mare care formează unul dintre aceste platouri. **(6p.)**
- Exemplu:** dacă fișierul `SIR.TXT` are conținutul alăturat,
- |                                           |
|-------------------------------------------|
| 10                                        |
| 11 211 211 211 328 400 400 1201 1201 1201 |
- atunci programul va afișa pe ecran **3 1201**.
- b) Descrieți succint, în limbaj natural, metoda utilizată la punctul a, justificând eficiența acesteia. **(4p.)**



**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Pentru rezolvarea cărei probleme dintre cele enumerate mai jos se poate utiliza metoda backtracking ? **(4p.)**
- a. determinarea reuniunii a 3 mulțimi                      b. determinarea tuturor divizorilor unui număr din 3 cifre
- c. determinarea tuturor elementelor mai mici decât 30000 din șirul lui Fibonacci                      d. determinarea tuturor variantelor în care se pot genera steagurile cu 3 culori (din mulțimea: "roșu", "galben", "albastru" și "alb"), având la mijloc culoarea "galben"

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Se consideră subprogramul recursiv definit alăturat. Ce valoare are expresia `bac(10,4)`? Care este cea mai mică valoare de 2 cifre a lui `u` pentru care funcția `bac(u,2)` are valoarea 1? **(6p.)**
- ```
int bac(int u, int x)
{if (u<x)return 0;
 if (x==u)return 1;
 if (u%x==0)return 0;
 return bac(u,x+1);
}
```
3. Să se scrie în limbajul C/C++ definiția completă a subprogramului `calcul`, care primește prin parametrul `n` un număr natural nenul de cel mult 9 cifre și furnizează prin parametrul `x` numărul obținut prin alăturarea cifrelor pare ale lui `n` considerate de la dreapta către stânga. Dacă `n` nu conține nicio cifră pară, `x` primește valoarea 0. **(10p.)**

**Exemplu:** în urma apelului `calcul(9278,x)`, `x` primește valoarea 82.

4. Fișierul text `NUMERE.TXT` conține pe prima linie un număr natural `n` ( $1 \leq n \leq 10000$ ) și pe a doua linie, `n` numere naturale cu cel mult 9 cifre fiecare, numere nu neapărat distincte. Aceste numere sunt dispuse în ordine **crescătoare** și separate între ele prin câte un spațiu.
- a) Scrieți un program C/C++ care citește valorile din fișier și, printr-o metodă eficientă din punct de vedere al timpului de executare și al spațiului de memorie utilizat, afișează pe ecran, cu un spațiu între ele, valoarea care apare de cele mai multe ori în fișier și de câte ori apare ea. Dacă există mai multe valori care apar de un număr maxim de ori, se va afișa cea mai mică dintre ele. **(6p.)**
- Exemplu:** dacă fișierul `NUMERE.TXT` are conținutul alăturat,
- ```
8
711 711 711 11111 11111 11111 191111 231111
```
- atunci programul va afișa pe ecran `711 3`.
- b) Descrieți succint, în limbaj natural, metoda utilizată la punctul a, justificând eficiența acesteia. **(4p.)**



**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Se generează în ordine crescătoare toate numerele de 4 cifre, cu cifre distincte, astfel încât diferența în valoare absolută dintre prima și ultima, respectiv a doua și a treia cifră este egală cu 2. Primele 11 soluții generate sunt, în ordine: 1023, 1203, 1243, 1423, 1463, 1573, 1643, 1683, 1753, 1793, 1863. Care dintre următoarele numere se va genera imediat înaintea numărului 9317? (4p.)
- a. 9247                      b. 9357                      c. 9207                      d. 8976

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Scrieți ce se va afișa în urma executării subprogramului alăturat, la apelul `F(57)`; (6p.)
- ```
void F(int x)
{
    if(x != 0)
    {
        F(x/2);
        cout << x%2; | printf("%d", x%2);
    }
}
```
3. a) Scrieți definiția completă a subprogramului **Ecuatie** care primește prin parametri **a**, **b** și **c** trei numere întregi,  $a \neq 0$ , de cel mult patru cifre fiecare, reprezentând coeficienții ecuației de gradul al II-lea:  $ax^2 + bx + c = 0$ . În funcție de soluțiile ecuației subprogramul va returna:
- cea mai mare dintre soluții dacă ecuația are două soluții reale distincte, dintre care cel puțin una pozitivă.
  - una dintre soluții dacă ecuația are două soluții egale și pozitive.
  - -32000 în celelalte cazuri.
- (10p.)
- b) Se consideră șirul **s**: 1, 1, 2, 1, 2, 3, 1, 2, 3, 4, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 6, 1, 2, ... . Pentru un număr natural **k**,  $0 < k \leq 10000$ , se cere să se determine valoarea elementului ce se află pe poziția **k** în șirul **s**.  
**Exemplu:** pentru **k=18** numărul cerut este 3.
- Scrieți un program C/C++ care citește de la tastatură valoarea numărului natural **k** și, prin apeluri utile ale funcției **Ecuatie**, determină valoarea elementului ce se află pe poziția **k** în șirul **s**, folosind un algoritm eficient din punctul de vedere al spațiului de memorie alocat și al timpului de executare. Valoarea astfel determinată se va scrie în fișierul text **sir.out**. (6p.)
- c) Descrieți succint, în limbaj natural, metoda de rezolvare folosită la punctul **b**, explicând în ce constă eficiența ei (3 – 4 rânduri) (4p.)



**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Se generează în ordine crescătoare toate numerele de 4 cifre, cu cifre distincte, astfel încât diferența în valoare absolută dintre ultimele două cifre ale fiecărui număr generat este egală cu 2. Primele opt soluții generate sunt, în ordine: 1024, 1035, 1042, 1046, 1053, 1057, 1064, 1068. Care dintre următoarele numere se va genera imediat după numărul 8975? **(4p.)**
- a. 8979                      b. 9013                      c. 8957                      d. 9024

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Ce se va afișa în urma executării subprogramului alăturat, la apelul `F(56);`? **(6p.)**
- ```
void F(int x)
{
    if(x)
    {
        F(x/2);
        cout << x%10; | printf("%d",x%10);
    }
}
```
3. **a)** Scrieți definiția completă a subprogramului **Cautare**, cu trei parametri, **n**, **x** și **v**, care primește prin parametrul **n** un număr natural ( $1 \leq n \leq 1000$ ), prin parametrul **x** un tablou unidimensional format din **n** componente (numere întregi de cel mult patru cifre fiecare:  $x_1, x_2, \dots, x_n$ ) memorate în ordine crescătoare și prin parametrul **v** un număr întreg de cel mult patru cifre, diferit de oricare dintre elementele tabloului unidimensional **x**. Subprogramul va căuta, în mod eficient din punct de vedere al timpului de executare, poziția pe care ar trebui inserată valoarea **v** în șirul **x** astfel încât să se obțină tot un șir ordonat crescător și returnează această poziție. **(6p.)**
- b)** Descrieți succint, în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența ei (3 – 4 rânduri). **(4p.)**
- c)** Fișierul text **sir.in** conține cel mult 1000 numere întregi de maximum patru cifre fiecare, numerele fiind diferite două câte două și despărțite prin câte un spațiu. Scrieți un program C/C++ care citește numerele din fișierul **sir.in** și, folosind apeluri utile ale subprogramului **Cautare**, construiește în memorie un tablou unidimensional care va conține toate numerele din fișierul **sir.in** ordonate crescător. Programul scrie în fișierul text **sir.out** șirul obținut, câte 10 elemente pe un rând, elementele de pe același rând fiind despărțite printr-un singur spațiu. **Exemplu:** dacă fișierul **sir.in** conține numerele: 7 -5 635 -456 0 8 587 -98 65 3 -8 atunci după executarea programului fișierul **sir.out** va conține:
- 456 -98 -8 -5 0 3 7 8 65 587
- 635
- (10p.)**

**Ministerul Educației, Cercetării și Inovării**  
**Centrul Național pentru Curriculum și Evaluare în Învățământul Preuniversitar**

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Prin metoda backtracking se generează toate anagramele (cuvintele obținute prin permutarea literelor) unui cuvânt dat. Știind că se aplică această metodă pentru cuvântul **solar**, precizați câte cuvinte se vor genera astfel încât prima și ultima literă din fiecare cuvânt generat să fie **vocală** (sunt considerate vocale caracterele **a, e, i, o, u**)? **(4p.)**
- a. 24                                      b. 6                                      c. 10                                      d. 12

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Se consideră funcția **suma**, definită alăturat. Ce valoare are **suma(3)**? Dar **suma(8)**? **(6p.)**
- ```
int Suma(int x)
{ if(x == 1) return 0;
  if(x%2==0) return Suma(x-1)+(x-1)*x;
  return Suma(x-1)-(x-1)*x;
}
```
3. **a)** Scrieți definiția completă a subprogramului **Cifra**, cu doi parametri, **n** și **x**, care primește prin intermediul parametrului **n** un număr natural de cel mult nouă cifre și furnizează prin parametrul **x** cea mai mare cifră a numărului transmis prin parametrul **n**. **(6p.)**
- b)** Scrieți un program **C/C++** care citește de la tastatură un număr natural **n**,  $n < 1000000000$ , și, utilizând apeluri utile ale subprogramului **Cifra**, afișează pe ecran mesajul **Da** în cazul în care numărul citit este format doar din cifre aparținând mulțimii  $\{0, 1, 2, 3, 4\}$  și afișează mesajul **Nu** în caz contrar. **(4p.)**
4. Se numește “număr mare” orice număr natural care are mai mult de nouă cifre.
- a)** Scrieți un program **C/C++** care citește de pe prima linie a fișierului text **NUMERE.IN** un număr natural **n** ( $10 \leq n < 1000$ ), iar de pe a doua linie **n** cifre despărțite prin câte un spațiu, dintre care cel puțin una nenulă, și afișează pe ecran cel mai mic “număr mare” format cu toate cele **n** cifre din fișier. Alegeți o metodă eficientă din punct de vedere al utilizării memoriei. **(6p.)**
- b)** Descrieți succint în limbaj natural metoda de rezolvare folosită explicând în ce constă eficiența ei (3 – 4 rânduri). **(4p.)**
- Exemplu:** dacă fișierul **NUMERE.IN** conține
- ```
10
7 9 4 0 9 0 1 1 8 8
```
- atunci se va afișa pe ecran “numărul mare” :
- ```
1001478899
```

**Subiectul III (30 de puncte)**

Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

1. Ce valoare are  $F(2758)$ , pentru funcția  $F$  definită alăturat? (4p.)

```
int F(int x)
{
    if(x == 0) return 0;
    if(x%10%2 == 0) return 2 + F(x/10);
    return 10 - F(x/10);
}
```

- a. 0                                      b. 20                                      c. 12                                      d. 4

Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.

2. Variabilele  $i$ ,  $j$  și  $aux$  sunt de tip întreg, iar înainte de prelucrare elementele tabloului unidimensional  $x$  sunt următoarele:  $x_0=10$ ,  $x_1=5$ ,  $x_2=-6$ ,  $x_3=7$ ,  $x_4=0$ ,  $x_5=-2$ .  
Ce valori se vor afișa în urma executării secvenței de program alăturate? (6p.)

```
for(int i = 0; i < 4; i++)
    for(int j = i + 1; j < 6; j++)
        if(x[i] > x[j])
        {
            aux = x[i];
            x[i] = x[j];
            x[j] = aux;
        }
for(i = 0; i < 6; i++)
    cout<<x[i]<<" ";
|   printf("%d ",x[i]);
```

3. a) Scrieți definiția completă a funcției **UltimaCifra** care primește prin cei doi parametri  $a$  și  $b$  câte un număr natural ( $0 < a < 30000$ ,  $0 < b < 30000$ ), calculează în mod eficient din punct de vedere al timpului de executare și returnează ultima cifră a numărului  $a^b$  ( $a$  la puterea  $b$ ). (6p.)

b) Descrieți succint, în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența ei (3 – 4 rânduri) (4p.)

c) Fișierul text **SIR.IN** conține pe prima sa linie un număr natural  $n$  ( $0 < n < 1001$ ), iar pe fiecare dintre următoarele  $n$  linii câte o pereche de numere naturale,  $x_i$   $y_i$  ( $1 \leq i \leq n$ ,  $x_i \leq 30000$ ,  $y_i \leq 30000$ ).

Scrieți programul C/C++ care citește numerele din fișierul **SIR.IN** și scrie în fișierul text

**SIR.OUT** ultima cifră a expresiei:  $X_1^{y_1} + X_2^{y_2} + \dots + X_n^{y_n}$ , folosind apeluri ale funcției **UltimaCifra**.

**Exemplu:** dacă fișierul **SIR.IN** are conținutul alăturat, atunci **SIR.OUT** va conține cifra 0. (10p.)

```
3
25 6
8 10
1 4589
```

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Dacă se utilizează metoda backtracking pentru a genera toate permutările de 4 obiecte și primele 5 permutări generate sunt, în această ordine, 4 3 2 1, 4 3 1 2, 4 2 3 1, 4 2 1 3, 4 1 3 2, atunci a 6-a permutare este: **(4p.)**
- a. 3 2 1 4                      b. 3 4 2 1                      c. 1 4 3 2                      d. 4 1 2 3

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Pentru definiția alăturată a subprogramului `sub`, scrieți ce valoare are `sub(3)`. Dar `sub(132764)`? **(6p.)**
- ```
long sub(long n)
{
    if (n!=0)
        if(n%2!=0) return n%10*sub(n/10);
        else return sub(n/10);
    else return 1;
}
```
3. Scrieți definiția completă a unui subprogram `s` cu trei parametri care primește prin intermediul parametrului `n` un număr natural de maximum 9 cifre, prin intermediul parametrului `c` o cifră și furnizează prin intermediul parametrului `k` numărul de cifre ale numărului `n` care aparțin intervalului  $[c-1, c+1]$ .  
**Exemplu:** pentru `n=1233` și `c=3`, `k` va avea valoarea 3, iar pentru `n=650` și `c=3`, `k` va avea valoarea 0. **(10p.)**
4. Fișierul `BAC.TXT` are pe prima linie două numere naturale `n` și `m` ( $0 < n < 1000$ ,  $0 < m < 1000$ ) separate prin câte un spațiu, pe linia a doua `n` numere întregi ordonate strict crescător, iar pe linia a treia `m` numere naturale distincte. Numerele din fișier aflate pe linia a doua și a treia au cel mult 6 cifre fiecare și sunt despărțite în cadrul liniei prin câte un spațiu. Să se scrie un program care citește toate numerele din fișier și afișează pe ecran, despărțite prin câte un spațiu, toate numerele de pe a doua linie a fișierului care apar și pe linia a treia a acestuia.  
**Exemplu:** dacă fișierul are următorul conținut:
- ```
6 5
2 3 4 5 8 9
4 5 2 11 8
```
- atunci se va afișa: 5 2 8 4, nu neapărat în această ordine.
- a) Descrieți în limbaj natural o metodă de rezolvare eficientă ca timp de executare. **(4p.)**
- b) Scrieți programul C/C++ corespunzător metodei descrise la punctul a). **(6p.)**

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. La un concurs participă 50 de sportivi împărțiți în 5 echipe, astfel încât în fiecare echipă să fie câte 10 sportivi. Problema determinării tuturor grupelor de câte 5 sportivi, câte unul din fiecare echipă, este similară cu generarea tuturor: **(4p.)**
- a. elementelor produsului cartezian  $A \times A \times A \times A \times A$ , unde  $A = \{1, 2, \dots, 10\}$       b. submulțimilor cu 5 elemente ale mulțimii  $\{1, 2, \dots, 10\}$
- c. permutărilor mulțimii  $\{1, 2, 3, 4, 5\}$       d. partițiilor mulțimii  $\{1, 2, \dots, 10\}$

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Pentru definiția alăturată a subprogramului `sub`, scrieți ce valoare are `sub(4)`. Dar `sub(132764)`? **(6p.)**
- ```
int sub(long n)
{
    if (n!=0)
        if (n%2!=0)
            return n%10+sub(n/10);
        else return sub(n/10);
    else return 0;
}
```
3. Scrieți un program C/C++ care citește de la tastatură două numere naturale nenule  $n$  și  $k$  ( $0 < n < 100$ ,  $0 < k < 4$ ) și apoi  $n$  numere reale pozitive care au cel mult 3 cifre la partea întreagă și cel mult 5 cifre zecimale și afișează pe ecran numărul de valori reale citite care au mai mult de  $k$  cifre zecimale.  
**Exemplu:** dacă pentru  $n$  se citește valoarea 5 și pentru  $k$  valoarea 2 și apoi șirul de numere reale 6.2 4.234 2 8.13 10.001 pe ecran se va afișa valoarea 2. **(10p.)**
4. Fișierul text **bac.in** conține pe prima sa linie un număr natural  $n$  ( $0 < n < 10000$ ), iar pe următoarea linie  $n$  numere naturale din intervalul  $[1, 100]$  separate prin câte un spațiu. Se cere să se citească din fișier toate numerele și să se afișeze pe ecran numărul sau numerele care apar de cele mai multe ori printre numerele citite de pe a doua linie a fișierului. Numerele afișate vor fi separate prin câte un spațiu. Alegeți un algoritm de rezolvare eficient atât din punctul de vedere al timpului de executare cât și al gestionării memoriei. .  
**Exemplu:** dacă fișierul **bac.in** are următorul conținut:
- ```
12
1 2 2 3 2 9 3 3 9 9 7 1
```
- pe ecran se vor afișa valorile 2, 3 și 9, nu neapărat în această ordine.
- a) Explicați în limbaj natural metoda utilizată justificând eficiența acesteia (4-6 rânduri) **(4p.)**
- b) Scrieți programul C/C++ ce rezolvă problema enunțată, corespunzător metodei descrise la punctul a). **(6p.)**

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Un program construiește și afișează elementele produsului cartezian  $A \times B \times C$  pentru mulțimile  $A = \{1, 2, 3, 4\}$ ,  $B = \{1, 2, 3\}$ ,  $C = \{1, 2\}$ . Care dintre următoarele triplete **NU** va fi afișat? **(4p.)**
- a.  $(3, 2, 1)$                       b.  $(1, 3, 2)$                       c.  $(1, 2, 3)$                       d.  $(2, 2, 2)$

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Pentru definiția alăturată a subprogramului `sub`, scrieți ce valoare are `sub(4)`. Dar `sub(123986)`? **(6p.)**
- ```
int sub(long n)
{
    if (n!=0)
        if (n%2!=0) return 1+sub(n/10);
        else return sub(n/10);
    else return 0;
}
```

3. a) Scrieți doar antetul unui subprogram `prim` cu doi parametri, care primește prin intermediul parametrului `n` un număr natural cu cel mult patru cifre și returnează prin intermediul parametrului `p` valoarea 1 dacă `n` este prim și 0 în caz contrar. **(2p.)**

b) Scrieți un program C/C++ care citește de la tastatură un număr natural `n` ( $3 < n < 10000$ ) și afișează pe ecran, despărțite prin câte un spațiu, primele `n` numerele prime, folosind apeluri utile ale subprogramului `prim`. **(8p.)**

**Exemplu:** pentru `n=4` pe ecran vor fi afișate numerele 2 3 5 7

4. Fișierul text `bac.in` conține pe prima sa linie un număr natural `n` ( $0 < n < 10000$ ), iar pe următoarea linie `n` numere naturale din intervalul  $[1, 100]$ . Se cere să se citească din fișier toate numerele și să se afișeze pe ecran, în ordine descrescătoare, toate numerele care apar pe a doua linie a fișierului și numărul de apariții ale fiecăruia. Dacă un număr apare de mai multe ori, el va fi afișat o singură dată. Fiecare pereche „valoare - număr de apariții” va fi afișată pe câte o linie a ecranului, numerele fiind separate printr-un spațiu, ca în exemplu. Alegeți un algoritm de rezolvare eficient din punctul de vedere al timpului de executare.

**Exemplu:** dacă fișierul `bac.in` are următorul conținut:

12

1 2 2 3 2 2 3 3 2 3 2 1

pe ecran se vor afișa, în această ordine, perechile:

3 4

2 6

1 2

- a) Explicați în limbaj natural metoda utilizată justificând eficiența acesteia (4-6 rânduri) **(4p.)**
- b) Scrieți programul C/C++ ce rezolvă problema enunțată, corespunzător metodei descrise la punctul a). **(6p.)**

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Problema generării tuturor codurilor formate din exact 4 cifre nenule, cu toate cifrele distincte două câte două, este similară cu generarea tuturor: **(4p.)**
- a. aranjamentelor de 9 elemente luate câte 4      b. permutărilor elementelor unei mulțimi cu 4 elemente
- c. elementelor produsului cartezian  $A \times A \times A \times A$  unde  $A$  este o mulțime cu 9 elemente      d. submulțimilor cu 4 elemente ale mulțimii  $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Pentru definiția alăturată a subprogramului  $f$ , scrieți ce valoare are  $f(8)$ . Dar  $f(1209986)$ ? **(6p.)**
- ```
int f(long x)
{
    int y, z;
    if (x==0) return x;
    else {y=x%10;
          z=f(x/10);
          if(y>z) return y;
          else return z;
    }
}
```
3. Scrieți definiția completă a subprogram  $\max$ , cu trei parametri,  $a$ ,  $b$ ,  $c$ , care primește prin intermediul parametrilor  $a$  și  $b$  două numere reale cu exact două cifre la partea întreagă și exact două zecimale fiecare. Subprogramul determină cel mai mare număr real dintre următoarele patru valori:  $a$ ,  $b$  și numerele reale obținute din  $a$  și  $b$  prin interschimbarea părții întregi cu partea fracționară în cadrul aceluiași număr. Această valoare este furnizată prin intermediul parametrului real  $c$ .  
**Exemplu:** dacă  $a=33.17$  și  $b=15.40$ ,  $c$  va avea valoarea  $40.15$  (cea mai mare valoare dintre  $33.17$ ,  $15.40$ ,  $17.33$  și  $40.15$ ) **(10p.)**
4. Se citește de pe prima linie a fișierului `numere.in` un număr natural  $n$  ( $0 < n < 10000$ ) și, de pe a doua linie a fișierului,  $n$  numere naturale din intervalul  $[1, 100]$  și se cere să se afișeze pe ecran, în ordine crescătoare, despărțite prin câte un spațiu, numărul sau numerele întregi din intervalul  $[1, 100]$  care **nu** apar printre numerele citite. Dacă pe a doua linie a fișierului apar toate numerele din intervalul precizat, se va afișa mesajul **NU LIPSESTE NICIUN NUMAR**. Alegeți un algoritm de rezolvare eficient din punctul de vedere al timpului de executare.  
**Exemplu:** pentru fișierul `numere.in` cu următorul conținut
- ```
12
4 2 3 1 6 5 7 8 9 11 10 100
```
- se vor afișa valorile `12 13 ... 99`.
- a) Explicați în limbaj natural metoda utilizată, justificând eficiența acesteia (4-6 rânduri). **(4p.)**
- b) Scrieți programul C/C++ ce rezolvă problema enunțată, corespunzător metodei descrise la punctul a). **(6p.)**

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. O clasă de 28 de elevi este la ora de educație fizică și profesorul dorește să formeze o echipă de 4 elevi. Ordinea elevilor în cadrul echipei nu are importanță. Algoritmul de generare a tuturor posibilităților de a forma o astfel de echipă este similar cu algoritmul de generare a tuturor: **(4p.)**
- a. aranjamentelor de 28 de elemente luate câte 4      b. combinațiilor de 28 de elemente luate câte 4
- c. partițiilor unei mulțimi cu 28 de elemente      d. elementelor produsului cartezian  $A \times A \times A \times A$ ,  $A$  fiind o mulțime cu 28 de elemente

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Pentru definiția alăturată a subprogramului `sub`, ce valoare are `sub(9)`? Dar `sub(132764)`? **(6p.)**
- ```
int sub(long n)
{
    if (n!=0)
        if (n%2!=0)
            return n%10+sub(n/10);
        else return sub(n/10);
    else return 0;
}
```
3. Fișierul text `bac.in` conține cel mult 1000 de numere naturale cu cel mult patru cifre fiecare, despărțite prin câte un spațiu. Scrieți programul C/C++ care citește numerele din fișier și afișează pe ecran, în ordine crescătoare, acele numere din fișier care au toate cifrele egale. Dacă fișierul nu conține niciun astfel de număr, se va afișa pe ecran mesajul **NU EXISTA**.
- Exemplu:** dacă fișierul `bac.in` conține numerele: 30 44 111 7 25 5 atunci pe ecran se va afișa 5 7 44 111. **(10p.)**
4. a) Scrieți definiția completă a unui subprogram `divi` cu doi parametri, care primește prin intermediul parametrului `n` un număr natural nenul cu cel mult 6 cifre și returnează prin intermediul parametrului `d` cel mai mic divizor propriu al lui `n` sau 0 în cazul în care `n` nu are niciun divizor propriu. De exemplu, pentru `n=15`, `d` va avea valoarea 3. **(4p.)**
- b) Scrieți programul C/C++ care citește de la tastatură un număr natural nenul `n` ( $1 < n < 1000000$ ) și afișează pe ecran cel mai mare număr natural mai mic decât `n` care îl divide pe `n`. Se vor folosi apeluri utile ale subprogramului `divi`.
- Exemple:** dacă `n=120`, se afișează pe ecran 60, iar dacă `n=43`, se afișează 1. **(6p.)**



**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Folosind cifrele  $\{1, 2, 3\}$  se generează, în ordinea crescătoare a valorii, toate numerele pare formate din trei cifre distincte. Astfel, se obțin în ordine, numerele: 132, 312. Folosind aceeași metodă, se generează numerele pare formate din patru cifre distincte din mulțimea  $\{1, 2, 3, 4\}$ . Care va fi al 4-lea număr generat ? (4p.)
- a. 2134                      b. 1432                      c. 2314                      d. 1423

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Pentru definiția alăturată a subprogramului **f**, scrieți ce valoare are **f(0,0)**. Dar **f(525,5)**? (6p.)
- |   |   |
|---|---|
| <pre>int f(int x,int y) { if(x==0)return 0;   else     if(x%10==y)       return f(x/10,y)+1;     else return f(x/10,y); }</pre> | <pre>int f(int x,int y) { if(x==0)return 0;   else     if(x%10==y)       return f(x/10,y)+1;     else return f(x/10,y); }</pre> |
|---|---|
3. Scrieți în C/C++ definiția completă a subprogramului **medie** care are doi parametri:
- **n**, prin care primește un număr natural ( $1 \leq n \leq 100$ ) ;
  - **v**, prin care primește un tablou unidimensional cu **n** elemente, numere naturale, fiecare element având cel mult patru cifre.
- Subprogramul returnează media aritmetică a elementelor din tablou. (10p.)
4. Fișierul text **NUMERE.IN** conține, pe mai multe linii, cel mult 30000 de numere naturale nenule mai mici sau egale cu 500, despărțite prin câte un spațiu.
- a) Scrieți programul C/C++ care, utilizând un algoritm eficient din punct de vedere al timpului de executare, afișează pe ecran, în ordine crescătoare, toate numerele care au apărut o singură dată din fișierul **NUMERE.IN**, despărțite prin câte un spațiu.
- Exemplu:** dacă fișierul **NUMERE.IN** conține numerele scrise alăturat, se vor afișa valorile următoare: 3 4 5 6 34 (6p.)
- |  |           |
|--|-----------|
|  | 2 23 34 3 |
|  | 8 9 9 23  |
|  | 6 8 9 2   |
|  | 4 5 23 9  |
- b) Descrieți succint, în limbaj natural, metoda de rezolvare folosită la punctul a), explicând în ce constă eficiența ei (3 – 4 rânduri). (4p.)

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Folosind cifrele  $\{2, 3, 4\}$  se generează, în ordinea crescătoare a valorii, toate numerele impare formate din trei cifre distincte. Astfel se obțin, în ordine, numerele: 243, 423. Folosind aceeași metodă, se generează numerele pare formate din patru cifre distincte din mulțimea  $\{2, 3, 4, 5\}$ . Care va fi al 5-lea număr generat? **(4p.)**
- a. 3452                      b. 3524                      c. 2534                      d. 3542

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Pentru definiția alăturată a subprogramului `f`, stabiliți ce valoare are `f(2)`. Dar `f(123)`? **(6p.)**
- ```
int f(int x)
{ if(x==0) return 0;
  else
    if(x%2==0) return 1+f(x/10);
    else return 2+f(x/10);
}
```
3. Scrieți în C/C++ definiția completă a subprogramului `suma` care are doi parametri:
- `n`, prin care primește un număr natural ( $1 \leq n \leq 100$ );
  - `v`, prin care primește un tablou unidimensional cu `n` elemente, numere întregi situate în intervalul  $[10, 30000]$ . Funcția returnează suma numerelor din tabloul `v` care au ultimele două cifre identice.
- Exemplu:** dacă `n=4` și `v=(123, 122, 423, 555)` funcția va returna 677 ( $=122+555$ ). **(10p.)**
4. Fișierul text `NUMERE.IN` conține, pe mai multe linii, cel mult 30000 de numere naturale nenule mai mici sau egale cu 500, numerele de pe fiecare linie fiind despărțite prin câte un spațiu. Fișierul conține cel puțin două numere distincte, fiecare având două cifre.
- a)** Scrieți programul C/C++ care citește toate numerele din fișierul `NUMERE.IN` și creează fișierul text `NUMERE.OUT` care să conțină pe prima linie cel mai mare număr de două cifre din fișierul `NUMERE.IN`, și de câte ori apare el în acest fișier, iar pe a doua linie, cel mai mic număr de două cifre din fișierul `NUMERE.IN` și de câte ori apare el în acest fișier. Alegeți o metodă de rezolvare eficientă din punct de vedere al memoriei utilizate și al timpului de executare. **(6p.)**
- b)** Descrieți succint, în limbaj natural, metoda de rezolvare folosită la punctul **a)**, explicând în ce constă eficiența ei (3 – 4 rânduri). **(4p.)**

**Exemplu:** dacă fișierul `NUMERE.IN` are conținutul alăturat:

|   |     |    |    |
|---|-----|----|----|
| 2 | 253 | 34 | 3  |
| 6 | 88  | 9  | 2  |
| 4 | 54  | 34 | 88 |

atunci fișierul `NUMERE.OUT` va avea următorul conținut:

|    |   |
|----|---|
| 88 | 2 |
| 34 | 2 |

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Folosind cifrele  $\{1, 2, 3\}$  se generează, în ordinea crescătoare a valorii, toate numerele formate din exact trei cifre, în care cifrele alăturate au valori consecutive. Astfel se obțin în ordine, numerele: 121, 123, 212, 232, 321 și 323. Folosind aceeași metodă se generează numere de patru cifre din mulțimea  $\{1, 2, 3, 4\}$  care îndeplinesc aceeași condiție. Care va fi al 5-lea număr generat ? (4p.)
- a. 2121                      b. 2123                      c. 3121                      d. 2323

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Pentru definiția alăturată a subprogramului `f`, stabiliți ce valoare are `f(2)`. Dar `f(123)`? (6p.)
- ```
int f(int x)
{ if(x==0) return 0;
  else
    if(x%2==0) return 3+f(x/10);
    else return 4+f(x/10);
}
```
3. Scrieți în C/C++ definiția completă a subprogramului `suma` care are doi parametri:
- `n`, prin care primește un număr natural ( $1 \leq n \leq 100$ );
  - `v`, prin care primește un tablou unidimensional cu `n` elemente, numere întregi, fiecare având exact trei cifre.
- Funcția returnează suma elementelor din tablou care au prima cifră egală cu ultima cifră. (10p.)
4. Fișierul text **NUMERE.IN** conține mai multe linii, pe fiecare linie existând câte un șir de numere naturale nenule mai mici sau egale decât 30000, despărțite prin câte un spațiu; fiecare linie se termină cu numărul 0 (care se consideră că nu face parte din șirul aflat pe linia respectivă) și conține cel puțin două valori.
- a) Scrieți programul C/C++ care afișează pe ecran valoarea maximă din șirul care conține cele mai puține numere. În cazul în care există mai multe șiruri cu același număr minim de numere, se va afișa cea mai mare valoare care apare în unul dintre aceste șiruri. Alegeți o metodă de rezolvare eficientă din punct de vedere al memoriei utilizate și al timpului de executare. (6p.)
- b) Descrieți succint, în limbaj natural, metoda de rezolvare folosită la punctul a), explicând în ce constă eficiența ei (3 – 4 rânduri). (4p.)

**Exemplu:** dacă fișierul **NUMERE.IN** are conținutul

2	253	34	3	0
6	88	9	3	0
4	54	88	12345	98 234 546 0

alăturat, atunci pe ecran se va afișa numărul 253.

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Folosind cifrele  $\{3, 4, 5\}$  se generează, în ordinea crescătoare a valorii, toate numerele impare formate din trei cifre distincte. Astfel se obțin, în ordine, numerele: 345, 435, 453, 543. Folosind aceeași metodă, se generează numerele impare formate din patru cifre distincte din mulțimea  $\{2, 3, 4, 5\}$ . Care va fi al 5-lea număr generat? **(4p.)**
- a. 3425                      b. 2534                      c. 4235                      d. 3245

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Pentru definiția alăturată a subprogramului `f`, stabiliți ce valoare are `f(1)`? Dar `f(100)`? **(6p.)**
- |  |  |
|--|--|
| <pre>int f(int x) { if(x==0) return 1;   else     return 1+f(x-1); }</pre> | <pre>int f(int x) { if(x==0) return 1;   else     return 1+f(x-1); }</pre> |
|--|--|
3. Scrieți programul C/C++ care citește de la tastatură un număr natural  $n$  ( $1 \leq n \leq 100$ ), apoi un șir de  $n$  numere întregi, cu cel mult 2 cifre fiecare, notat  $a_1, a_2, a_3, \dots, a_n$ , apoi un al doilea șir de  $n$  numere întregi, cu cel mult 2 cifre fiecare, notat  $b_1, b_2, b_3, \dots, b_n$ . Fiecare șir conține atât valori pare, cât și impare. Programul afișează pe ecran suma acelor numere impare din șirul  $b$  care sunt mai mici decât suma tuturor numerelor pare din șirul  $a$ .  
**Exemplu:** pentru  $n=4$  și numerele 2, 3, 7, 8 respectiv 4, 3, 1, 8 se afișează valoarea 4 pentru că numerele 3 și 1 sunt mai mici decât suma numerelor pare din șirul  $a$ , care este 10. **(10p.)**
4. Se consideră subprogramul `CMMMC` care primește prin cei doi parametri,  $x$  și  $y$ , două numere naturale ( $1 \leq x \leq 10000$ ,  $1 \leq y \leq 10000$ ) și returnează cel mai mic multiplu comun al lor.
- a) Scrieți numai antetul subprogramului `CMMMC`. **(4p.)**
- b) Fișierul text `NUMERE.IN` conține, pe fiecare linie, câte două numere naturale nenule mai mici sau egale decât 10000, despărțite printr-un spațiu. Scrieți un program C/C++ care, pentru fiecare linie  $k$  din fișierul `NUMERE.IN`, citește cele două numere de pe această linie și scrie în fișierul text `NUMERE.OUT`, tot pe linia  $k$ , cel mai mic multiplu comun al acestora, ca în exemplu.
- Se vor utiliza apeluri utile ale subprogramului `CMMMC`. **(6p.)**
- |   |                            |   |                     |
|---|----------------------------|---|---------------------|
| <b>Exemplu:</b> dacă fișierul <code>NUMERE.IN</code> are conținutul alăturat: | <pre>12 14 11 12 4 8</pre> | atunci fișierul <code>NUMERE.OUT</code> va avea următorul conținut: | <pre>84 132 8</pre> |
|---|----------------------------|---|---------------------|

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Folosind cifrele  $\{1, 2, 3\}$  se generează, în ordinea crescătoare a valorii, toate numerele impare formate din trei cifre distincte. Astfel se obțin, în ordine, numerele: 123, 213, 231, 321. Folosind aceeași metodă, se generează numerele impare formate din patru cifre distincte din mulțimea  $\{1, 2, 3, 4\}$ . Care va fi al 5-lea număr generat ? **(4p.)**
- a. 2413                      b. 1423                      c. 2431                      d. 3241

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Pentru definiția alăturată a subprogramului **f**, scrieți ce valoare are **f(51)**. Dar **f(100)**? **(6p.)**
- ```
int f(int x)
{ if(x==50) return 1;
  else
    return 2+f(x-1);
}
```
3. Scrieți programul C/C++ care citește de la tastatură un număr natural **n** ( $1 \leq n \leq 100$ ), apoi un șir de **n** numere întregi, cu cel mult 2 cifre fiecare, notat  $a_1, a_2, a_3, \dots, a_n$ , apoi un al doilea șir de **n** numere întregi, cu cel mult 2 cifre fiecare, notat  $b_1, b_2, b_3, \dots, b_n$ . Fiecare șir conține atât valori pare, cât și impare. Programul afișează pe ecran suma acelor numere din șirul **b** care sunt strict mai mici decât media aritmetică a tuturor numerelor pare din șirul **a**.  
**Exemplu:** pentru **n=4** și numerele 2, 3, 7, 8 respectiv 44, 3, 1, 8 se afișează valoarea 4 pentru că numerele 3 și 1 sunt mai mici decât media aritmetică a numerelor pare din șirul **a**, care este 5. **(10p.)**
4. Se consideră subprogramul **CMMDC** care primește prin cei doi parametri, **x** și **y**, două numere naturale ( $1 \leq x \leq 10000$ ,  $1 \leq y \leq 10000$ ) și returnează cel mai mare divizor comun al lor.

**a) Scrieți definiția completă a subprogramului CMMDC. (4p.)**

**b) Fișierul text NUMERE.IN** conține, pe fiecare linie, câte două numere naturale nenule mai mici sau egale decât 10000, despărțite printr-un spațiu, reprezentând numitorul și numărătorul câte unei fracții. Scrieți un program C/C++ care, pentru fiecare linie **k** din fișierul **NUMERE.IN**, citește numitorul și numărătorul fracției de pe această linie și scrie în fișierul text **NUMERE.OUT**, tot pe linia **k**, numitorul și numărătorul acestei fracții, adusă la forma ireductibilă, ca în exemplu. Se vor utiliza apeluri utile ale subprogramului **CMMDC**. **(6p.)**

|                                 |       |                           |       |
|---------------------------------|-------|---------------------------|-------|
| <b>Exemplu:</b> dacă fișierul   | 12 14 | atunci fișierul           | 6 7   |
| <b>NUMERE.IN</b> are conținutul | 11 12 | <b>NUMERE.OUT</b> va avea | 11 12 |
| alăturat:                       | 2 2   | următorul conținut:       | 1 1   |
|                                 | 4 8   |                           | 1 2   |

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. La examenul de bacalaureat, un elev primește un test format dintr-un subiect de tip I, unul de tip II și unul de tip III. Știind că pentru fiecare tip de subiect sunt elaborate exact 100 de variante, algoritmul de generare a tuturor posibilităților de a forma un test este similar cu algoritmul de generare a:
- (4p.)**
- a. elementelor produsului cartezian                      b. aranjamentelor  
c. permutărilor                                                      d. submulțimilor

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Se consideră subprogramul `f`, definit alăturat. Ce se afișează la apelul `f(4)`? **(6p.)**
- ```
void f(int n)
{
    cout<<"*"; | printf("*");
    if(n>2)
    {
        f(n-1);
        cout<<"#"; | printf("#");
    }
}
```
3. Scrieți definiția completă a subprogramului `numar`, cu trei parametri, care primește prin intermediul parametrului `n` un număr natural format din cel mult 9 cifre, iar prin intermediul parametrilor `c1` și `c2` câte o cifră nenulă; subprogramul returnează numărul obținut prin înlocuirea în numărul primit prin parametrul `n` a fiecărei apariții a cifrei `c1` cu cifra `c2`. Dacă `c1` nu apare în `n`, subprogramul returnează valoarea `n`.  
**Exemplu:** pentru `n=12445`, `c1=4` și `c2=7` valoarea returnată va fi 12775. **(10p.)**
4. Fișierul text `bac.txt` conține cel puțin două și cel mult 1000 de numere naturale distincte, dintre care cel puțin două sunt pare. Numerele sunt separate prin câte un spațiu și fiecare dintre ele are cel mult 9 cifre.
- a) Scrieți un program C/C++ care determină cele mai mari două numere pare din fișier, utilizând un algoritm eficient din punct de vedere al timpului de executare și al spațiului de memorie utilizat. Cele două numere vor fi afișate pe ecran, în ordine descrescătoare, separate printr-un spațiu.  
**Exemplu:** dacă fișierul conține numerele: 5123 8 6 12 3 se va afișa: 12 8 **(6p.)**
- b) Descrieți succint, în limbaj natural, algoritmul utilizat, justificând eficiența acestuia. **(4p.)**













**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Având la dispoziție cifrele 0, 1 și 2 se pot genera, în ordine crescătoare, numere care au suma cifrelor egală cu 2. Astfel, primele 6 soluții sunt 2, 11, 20, 101, 110, 200. Folosind același algoritm, se generează numere cu cifrele 0, 1, 2 și 3 care au suma cifrelor egală cu 4. Care va fi al 7-lea număr din această generare? **(4p.)**
- a. 130                      b. 301                      c. 220                      d. 103

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Se consideră subprogramul `f` definit alăturat. Ce se va afișa în urma apelului `f(14);`? **(6p.)**
- ```
void f(int x)
{
    if (x<=10)
        cout<<0<<" ";|printf("%d ",0);
    else
        {f(x-2);
         cout<<x<<" ";|printf("%d ",x);
        }
}
```
3. Subprogramul `ordonare` primește prin parametrul `x` un tablou unidimensional cu cel mult 100 de elemente numere reale, iar prin parametrul `n` un număr întreg ce reprezintă numărul efectiv de elemente ale tabloului `x`. Subprogramul ordonează crescător elementele tabloului și furnizează, tot prin intermediul parametrului `x`, tabloul ordonat.
- a) Scrieți numai antetul acestui subprogram. **(4p.)**
- b) Scrieți un program C/C++ care citește de la tastatură două numere naturale, `n` și `m` ( $1 \leq n \leq 100$  și  $m \leq n$ ), și apoi un șir de `n` numere reale distincte. Folosind apeluri utile ale subprogramului `ordonare`, programul afișează pe prima linie a ecranului, cele mai mari `m` elemente din șirul citit (în ordine crescătoare a valorilor lor), iar pe a doua linie de ecran, cele mai mici `m` elemente din șir (în ordine descrescătoare a valorilor lor). Numerele afișate pe aceeași linie vor fi separate prin câte un spațiu. **(10p.)**
- Exemplu :** dacă `n=9`, `m=3`, iar șirul este (14.2, 60, -7.5, -22, 33.8, 80, 4, 10, 3) se va afișa pe ecran:
- ```
33.8 60 80
3 -7.5 -22
```
4. Scrieți un program C/C++ care creează fișierul text `SIR.TXT` și scrie în el toate șirurile formate din două caractere distincte, litere mari ale alfabetului englez, astfel încât niciun șir să nu fie format din două vocale alăturate. Fiecare șir va fi scris pe câte o linie a fișierului. **(6p.)**

**Subiectul III (30 de puncte)**

Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

1. În câte dintre permutările elementelor mulțimii  $\{ 'I', 'N', 'F', 'O' \}$  vocala 'I' apare pe prima poziție? (4p.)
- a. 1                      b. 24                      c. 6                      d. 12

Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.

2. Subprogramul **f** este definit alăturat.

Ce se afișează ca urmare a apelului **f(1,3)**? (6p.)

```
void f (int x,int y)
{   if (x<y){x=x+1;f(x,y);
      y=y-1;f(x,y);
    }
    else
      cout<<x<<y; printf("%d%d",x,y);
}
```

3. Scrieți programul C/C++ care citește de la tastatură un număr natural  $n$  ( $1 \leq n \leq 99$ ), impar, și construiește în memorie un tablou unidimensional  $A=(A_1, A_2, \dots, A_n)$  cu elementele mulțimii  $\{1, 2, \dots, n\}$  astfel încât elementele de pe poziții impare formează șirul crescător  $1, 2, \dots, [(n+1)/2]$ , iar elementele de pe poziții pare șirul descrescător  $n, n-1, \dots, [(n+1)/2]+1$ .

**Exemplu:** pentru  $n=11$  se va construi tabloul **A** :

1	11	2	10	3	9	4	8	5	7	6
---	----	---	----	---	---	---	---	---	---	---

Programul va crea un fișier text **TABLOU.TXT**. Elementele tabloului se vor scrie, în ordine, pe prima linie a fișierului, cu câte un spațiu între ele. (10p.)

4. a) Scrieți definiția completă a subprogramului **dcm**, cu doi parametri, care:
- primește prin parametri **a** și **b** două valori naturale din intervalul  $[1, 30000]$
  - returnează o valoare naturală reprezentând cel mai mare număr care este atât divizor al lui **a** cât și divizor al lui **b**.

**Exemplu:** dacă  $a=100$  și  $b=120$ , subprogramul returnează valoarea 20. (6p.)

b) Scrieți un program C/C++ care citește de la tastatură două numere naturale **a** și **b**, numere din intervalul  $[1, 30000]$  și determină, folosind apeluri utile ale subprogramului **dcm**, cel mai mare număr care este divizor al fiecăruia dintre numerele **a** și **b** și are proprietatea că este un produs de două sau mai multe numere prime distincte. Programul afișează pe ecran numărul cu proprietatea cerută, iar dacă nu există un astfel de număr, afișează mesajul **nu exista**.

**Exemplu:** dacă  $a=60$  și  $b=72$ , atunci se afișează 6

iar dacă  $a=100$  și  $b=75$ , atunci se afișează **nu exista**. (4p.)

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Un elev realizează un program care citește o valoare naturală pentru o variabilă  $n$  și apoi afișează în fișierul `permut.txt`, pe prima linie, valoarea lui  $n$ , apoi toate permutările mulțimii  $\{1, 2, \dots, n\}$ , câte o permutare pe câte o linie a fișierului. Rulând programul pentru  $n=3$ , fișierul va conține cele 7 linii alăturate. Dacă va rula din nou programul pentru  $n=4$ , ce va conține a 8-a linie din fișier? (4p.)
- |     |   |
|-----|---|
| 3   | 3 |
| 3 2 | 1 |
| 3 1 | 2 |
| 2 3 | 1 |
| 2 1 | 3 |
| 1 3 | 2 |
| 1 2 | 3 |
- a. 2134                      b. 2143                      c. 3421                      d. 3412

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Funcția recursivă  $f$  este astfel definită încât  $f(1)=8$ , iar  $f(n+1)=2*f(n)-4$  pentru orice  $n$  natural nenul.
- a) Ce valoare are  $f(5)$  ? (3p.)
- b) Care este cea mai mare valoare pe care o poate lua  $x$  astfel încât  $f(x) < 1000$  ? (3p.)
3. Scrieți definiția completă a funcției  $f$ , care primește prin intermediul parametrului  $n$  un număr natural nenul ( $2 \leq n \leq 200$ ), iar prin intermediul parametrului  $a$  un tablou unidimensional care conține  $n$  valori întregi, fiecare dintre aceste valori întregi având cel mult patru cifre. Funcția returnează valoarea 1 dacă elementele tabloului formează un șir crescător, valoarea 2 dacă elementele tabloului formează un șir descrescător, valoarea 0 dacă elementele tabloului formează un șir constant și valoarea -1 în rest. (10p.)
4. Fișierul text `număr.txt` conține pe prima linie o valoare naturală  $n$  cu exact 9 cifre nenule distincte. Scrieți un program eficient din punctul de vedere al timpului de executare care citește din fișier numărul  $n$  și afișează pe ecran cea mai mică valoare  $m$  formată din exact aceleași cifre ca și  $n$ , astfel încât  $m > n$ . În cazul în care nu există o astfel de valoare, programul va afișa pe ecran mesajul **Nu exista**.
- Exemplu:** Dacă fișierul `număr.txt` conține numărul 257869431, se va afișa pe ecran numărul 257891346.
- a) Descrieți succint, în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența ei (3 – 4 rânduri). (4p.)
- b) Scrieți un program C/C++ care rezolvă problema conform metodei descrise. (6p.)

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Un program citește o valoare naturală nenulă pentru  $n$  și apoi generează și afișează, în ordine crescătoare lexicografic, toate combinațiile formate din  $n$  cifre care aparțin mulțimii  $\{0,1\}$ . Astfel, pentru  $n=2$ , combinațiile sunt afișate în următoarea ordine: 00, 01, 10, 11. Dacă se rulează acest program și se citește pentru  $n$  valoarea 9, imediat după combinația 011011011 va fi afișată combinația: (4p.)
- a. 011100100      b. 011011100      c. 011011011      d. 011100000

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Funcția  $f$  are definiția alăturată. Scrieți cinci valori de apel pe care le poate avea  $n$  astfel încât, pentru cele 5 apeluri corespunzătoare acestor valori, să se obțină 5 valori ale funcției, distincte două câte două. (6p.)
- ```
int f(int n)
{
    if (n<=9) return 0;
    if (n%5==0) return 0;
    return 1+f(n-3);
}
```
3. Funcția  $f$  primește prin intermediul parametrului  $n$  un număr natural nenul ( $2 \leq n \leq 200$ ), iar prin intermediul parametrului  $a$  un tablou unidimensional care conține  $n$  valori întregi nenule (fiecare dintre aceste valori întregi având cel mult patru cifre). Funcția returnează valoarea  $-1$  dacă numărul de valori negative din tabloul  $a$  este strict mai mare decât numărul de valori pozitive din tablou, valoarea  $0$  dacă numărul de valori negative din  $a$  este egal cu numărul de valori pozitive din tablou și valoarea  $1$  dacă numărul de valori pozitive din tabloul  $a$  este strict mai mare decât numărul de valori negative din  $a$ . Scrieți definiția completă a funcției  $f$ . (10p.)
4. a) Scrieți un program C/C++ care citește de la tastatură un număr natural nenul,  $s$ , având maximum 9 cifre, și printr-o metodă eficientă din punct de vedere al timpului de executare, determină și scrie în fișierul `rez.dat` trei valori naturale a căror sumă este egală cu  $s$ , și al căror produs este maxim. Cele trei valori vor fi scrise în ordine crescătoare pe prima linie a fișierului `rez.dat`, separate prin câte un spațiu. (6p.)
- Exemplu:** dacă se citește valoarea 5, fișierul `rez.dat` va avea o linie cu conținutul 1 2 2.
- b) Descrieți succint, în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența ei (3 – 4 rânduri). (4p.)

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Un program citește o valoare naturală nenulă pentru  $n$  și apoi generează și afișează, în ordine descrescătoare lexicografic, toate combinațiile de  $n$  cifre care aparțin mulțimii  $\{0,1\}$ . Astfel, pentru  $n=2$ , combinațiile sunt afișate în următoarea ordine: 11, 10, 01, 00. Dacă se rulează acest program și se citește pentru  $n$  valoarea 8, imediat după combinația 10101000 va fi afișată combinația: (4p.)
- a. 01010111                      b. 10100111                      c. 10101001                      d. 10100100

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Funcția **f** are definiția alăturată. Scrieți 4 valori de apel pe care le poate avea  $n$  astfel încât, pentru cele 4 apeluri, corespunzătoare acestor valori, să se obțină 4 valori, distincte două câte două. **(6p.)**
- ```
int f(int n)
{
    if (n<=9) return 0;
    if (n%4==0) return 0;
    return 1+f(n-3);
}
```
3. Funcția **verif** primește prin intermediul a trei parametri, notați **a**, **b** și **c**, trei valori naturale nenule, fiecare de maximum patru cifre. Funcția returnează valoarea 1 dacă cele trei valori pot constitui laturile unui triunghi și valoarea 0 în caz contrar.
- a) Scrieți definiția completă a funcției **verif**. (5p.)
- b) Scrieți un program **C/C++** care citește de la tastatură șase valori naturale nenule, fiecare de maximum patru cifre, apoi verifică, utilizând apeluri utile ale funcției **verif**, dacă primele trei numere citite pot constitui laturile unui triunghi și dacă ultimele trei numere citite pot constitui laturile unui triunghi; în caz afirmativ, programul afișează pe ecran mesajul **congruente** dacă cele două triunghiuri sunt congruente sau mesajul **necongruente** dacă cele două triunghiuri nu sunt congruente; dacă cel puțin unul dintre cele două triplete de valori nu pot constitui laturile unui triunghi, programul va afișa pe ecran mesajul **nu**. **(5p.)**
4. Fișierul **BAC.DAT** conține pe prima linie, separate printr-un spațiu, două valori naturale  $n$  și  $m$  ( $2 \leq n \leq 1000$ ,  $2 \leq m \leq 1000$ ), pe a doua linie  $n$  valori întregi și pe a treia linie  $m$  valori întregi. Valorile de pe a doua și de pe a treia linie apar în fișier în ordine strict crescătoare, sunt separate prin câte un spațiu și au cel mult 4 cifre fiecare.
- Se cere afișarea pe ecran a două valori, dintre cele aflate în poziții consecutive pe a treia linie a fișierului, care determină intervalul închis în care se află un număr maxim de valori de pe a doua linie a fișierului. Se va utiliza o metodă eficientă din punct de vedere al timpului de executare și al spațiului de memorie utilizat. Se garantează că cel puțin un număr aflat pe a doua linie a fișierului aparține unuia dintre intervalele determinate de numerele de pe a treia linie a fișierului.
- Exemplu:** dacă fișierul **BAC.DAT** are conținutul
- alăturat, programul va afișa: 1 9

10 4

-1 1 3 4 5 6 10 15 16 117

0 1 9 20
- Explicație: cele patru numere de pe a treia linie a fișierului determină trei intervale:  $[0,1]$ ,  $[1,9]$ ,  $[9,20]$ ; în intervalul  $[1,9]$  se află 5 valori de pe a doua linie a fișierului, acesta fiind numărul maxim de valori aflate în unul dintre cele trei intervale.
- a) Descrieți succint, în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența ei (3 – 4 rânduri). (4p.)
- b) Scrieți un program **C/C++** care să rezolve problema conform metodei descrise. (6p.)



**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Subprogramul recursiv alăturat este definit incomplet.  
Care dintre următoarele expresii poate înlocui punctele de suspensie astfel încât, în urma apelului, subprogramul  $f$  să returneze suma primelor două cifre ale numărului primit prin intermediul parametrului  $x$ ?  
**Exemplu:** în urma apelului  $f(2318)$  valoarea returnată este 5. **(4p.)**

```
int f(int x){  
    if (...)  
        return x%10 + x/10;  
    else  
        return f(x/10);  
}
```

- a.  $x \leq 100$                       b.  $x \leq 99$                       c.  $x == 99$                       d.  $x != 0$

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Se generează, utilizând metoda backtracking, cuvintele cu exact 3 litere din mulțimea  $\{a, x, c, f, g\}$ . Dacă primele patru cuvinte generate sunt, în ordine, **aaa, aax, aac, aaf**, scrieți ultimele trei cuvinte care încep cu litera **a**, în ordinea în care vor fi generate. **(6p.)**

3. Tabloul unidimensional  $v$ , declarat global, memorează exact 50 de numere întregi:

$v_1, v_2, \dots, v_{50}$ .

Subprogramul **Calcul** primește prin intermediul parametrului  $k$  un număr natural nenul ( $k \leq 50$ ) și furnizează prin intermediul parametrului  $s$  suma tuturor elementelor pozitive, din tabloul  $v$ , cu indici mai mari sau egali cu  $k$  sau 0 dacă toate elementele menționate sunt negative.

- a) Scrieți doar antetul subprogramului **Calcul**. **(2p.)**

b) Scrieți un program în limbajul C/C++ care citește de la tastatură cele 50 de componente întregi ale tabloului  $v$  și două numere naturale nenule  $x$  și  $y$  ( $x < y \leq 50$ ). Programul afișează suma elementelor pozitive din tablou, cu indici cuprinși între  $x$  și  $y$  inclusiv, sau 0 dacă toate elementele menționate sunt negative, folosind apeluri utile la subprogramul **Calcul**. **(8p.)**

4. Pe prima linie a fișierului text **DATE.TXT** se găsește o valoare naturală  $k$  ( $k \leq 1000000$ ).

a) Scrieți un program C/C++ care citește din fișierul **DATE.TXT** valoarea  $k$  și afișează, pe ecran, toate perechile de numere naturale nenule  $x, y$  ( $x \leq y$ ) cu proprietatea că  $x^2 + y^2 = k$ . Fiecare pereche va fi afișată pe câte o linie, numerele fiind despărțite printr-un spațiu. Alegeți o metodă de rezolvare eficientă din punctul de vedere al timpului de executare.

**Exemplu:** dacă fișierul **DATE.TXT** conține numărul 1000000, pe ecran se vor afișa, nu neapărat în această ordine, perechile alăturate. **(6p.)**

280	960
352	936
600	800

- b) Descrieți succint, în limbaj natural, metoda utilizată, justificând eficiența acesteia **(4p.)**

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Se consideră subprogramul recursiv alăturat, definit incomplet.  
Cu ce valoare trebuie înlocuite punctele de suspensie, pentru ca funcția să returneze cifra minimă a numărului natural nenul transmis prin intermediul parametrului  $x$ ?

(4p.)

```
int Min(int x){  
    int c;  
    if (x==0) return ...;  
    else {  
        c=Min(x/10);  
        if (c < x%10) return c;  
        else return x%10;  
    }  
}
```

a. -1

b. 1

c. 9

d. 0

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Utilizând metoda backtracking se generează toate submulțimile nevide ale mulțimii  $\{3, 6, 2, 5\}$ . Primele șase submulțimi generate sunt, în ordine:  $\{3\}$ ,  $\{3, 6\}$ ,  $\{3, 6, 2\}$ ,  $\{3, 6, 2, 5\}$ ,  $\{3, 6, 5\}$ ,  $\{3, 2\}$ . Care sunt, în ordinea obținerii, ultimele trei submulțimi, generate după această regulă? (6p.)

3. Scrieți în limbajul C/C++ definiția completă a subprogramului **numar**, cu exact doi parametri, care primește prin intermediul parametrului  $x$  un număr natural nenul de cel mult 2 cifre, și prin intermediul parametrului  $y$  un număr natural nenul de cel mult 9 cifre. Subprogramul returnează cel mai mare număr natural  $z$  pentru care există un număr natural  $k$  astfel încât  $z = x^k$  și  $z \leq y$ .

**Exemplu:** pentru  $y=18$  și  $x=2$  subprogramul va returna valoarea  $16 (=2^4 < 18)$ . (10p.)

4. Pe prima linie a fișierului text **DATE.TXT** se află două numere naturale nenule  $n$  și  $m$  ( $n \leq 3000$ ,  $m \leq 3000$ ), pe a doua linie un șir de  $n$  numere naturale, ordonate crescător, având fiecare cel mult 9 cifre, iar pe linia a treia un șir de  $m$  numere naturale, ordonate descrescător, având fiecare cel mult 9 cifre. Numerele sunt despărțite, în cadrul liniilor, prin câte un spațiu.

**a)** Scrieți programul C/C++ care citește numerele din fișier și afișează, pe ecran, doar numerele pare din cele două șiruri, ordonate crescător. Programul nu va afișa nimic dacă nu există numere pare în cele două șiruri. Alegeți o metodă de rezolvare eficientă ca timp de executare.

**Exemplu:** dacă fișierul are conținutul alăturat, pe ecran se va afișa: 2 4 4 32 42 42 88 88 (6p.)

**b)** Descrieți succint, în limbaj natural, metoda utilizată, justificând eficiența acesteia. (4p.)

```
5 8  
2 4 7 37 42  
88 88 67 45 42 32 4 1
```

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Se consideră subprogramul recursiv alăturat,  $s$ , definit incomplet.  
Cu ce expresie pot fi înlocuite punctele de suspensie astfel încât, în urma apelului  $s(2)$ , să se afișeze 3 caractere  $*$ ? (4p.)

```
void s(int x)
{ cout<<'*';
  if (...) {
    cout<<'*';
    s(x-1);
  }
}
```

a.  $x>1$

b.  $x>2$

c.  $x\geq 3$

d.  $x>0$

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Se utilizează metoda backtracking pentru a genera toate cuvintele formate din două litere distincte din mulțimea  $\{w, x, z, y\}$  astfel încât niciun cuvânt să nu înceapă cu litera  $x$  și niciun cuvânt să nu conțină litera  $w$  lângă litera  $z$ . Cuvintele vor fi generate în ordinea  $wx$ ,  $wy$ ,  $zx$ ,  $zy$ ,  $yw$ ,  $yx$ ,  $yz$ . Folosind aceeași metodă se generează toate cuvintele de două litere distincte din mulțimea  $\{w, x, z, y, t\}$  astfel încât niciun cuvânt să nu înceapă cu litera  $x$  și niciun cuvânt să nu conțină litera  $w$  lângă litera  $z$ . Care sunt a treia și a patra soluție generată? (6p.)

3. Subprogramul  $Nr$  are un singur parametru,  $k$ , prin intermediul căruia primește un număr natural de cel puțin 3 cifre și cel mult 9 cifre, cu toate cifrele nenule. Subprogramul furnizează tot prin intermediul parametrului  $k$ , valoarea obținută prin eliminarea primei și ultimei cifre a numărului transmis la apel.

**Exemplu:** dacă subprogramul primește prin intermediul parametrului  $k$  valoarea 12438, în urma apelului subprogramului  $Nr$ ,  $k$  va primi valoarea 243.

Scrieți, în limbajul C/C++, definiția completă a subprogramului  $Nr$ .

(10p.)

4. Pe prima linie a fișierului text **DATE.TXT** se află un șir de cel mult 10000 de numere întregi, având cel mult 4 cifre fiecare. Numerele sunt despărțite prin câte un spațiu.

a) Scrieți un program C/C++ care citește numerele din fișier și afișează pe ecran lungimea maximă a unei secvențe de numere din șir, cu proprietatea că oricare două numere din secvență, aflate pe poziții consecutive, au parități diferite. Pe a doua linie a ecranului, programul va afișa o secvență de lungime maximă, valorile fiind despărțite prin câte un spațiu. Dacă există mai multe secvențe de lungime maximă, se va afișa una dintre ele, oricare. Alegeți o metodă de rezolvare eficientă ca timp de executare.

**Exemplu:** dacă fișierul conține, în ordine, numerele 2 4 3 2 7 4 6 2 7 8 12, se va afișa:

5

4 3 2 7 4

(6p.)

b) Descrieți succint, în limbaj natural, metoda utilizată, justificând eficiența acesteia. (4p.)

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Aplicând metoda backtracking pentru a genera toate permutările celor  $n$  elemente ale unei mulțimi, o soluție se memorează sub forma unui tablou unidimensional  $x_1, x_2, \dots, x_n$ . Dacă sunt deja generate valori pentru componentele  $x_1, x_2, \dots, x_{k-1}$ , iar pentru componenta curentă,  $x_k$  ( $1 < k < n$ ), a fost găsită o valoare convenabilă, atunci se încearcă alegerea **(4p.)**
- a. unei noi valori pentru componenta  $x_{k-1}$       b. unei valori pentru componenta  $x_{k+1}$   
c. unei noi valori pentru componenta  $x_k$       d. unei noi valori pentru componenta  $x_1$

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Ce afișează subprogramul **F**, descris alăturat, la apelul **F(5)**? **(6p.)**

```
void F(int x)
{
    cout<<x; | printf("%d",x);
    if(x>=3)
        F(x-2);
    cout<<x-1; | printf("%d",x-1);
}
```

3. Scrieți definiția completă a subprogramului **divizor**, cu trei parametri, prin care primește 3 numere naturale nenule cu cel mult 9 cifre fiecare și returnează numărul divizorilor comuni tuturor celor 3 numere.

**Exemplu:** dacă numerele primite ca parametri sunt 24, 20 și 12, subprogramul returnează valoarea 3 (divizorii comuni sunt 1, 2 și 4). **(10p.)**

4. Se consideră un șir **s** format după regula alăturată, unde s-a notat cu  $a \Theta b$  numărul obținut prin concatenarea cifrelor lui **a** și **b**, în această ordine.

$$s_n = \begin{cases} x & \text{dacă } n=1 \\ x+1 & \text{dacă } n=2 \\ s_{n-1} \Theta s_{n-2} & \text{dacă } n>2 \end{cases}$$

**Exemplu:** pentru  $x=2$  se obține șirul:

2, 3, 32, 323, 32332, ....

Fișierul text **SIR.TXT** conține pe prima linie două numere,  $x$  ( $1 \leq x \leq 20$ ) și  $k$  ( $1 \leq k \leq 5000$ ), separate printr-un spațiu, iar pe a doua linie un număr format din exact  $k$  cifre, reprezentând un termen al șirului **s** (diferit de  $x$ ). Cifrele numărului **nu** sunt separate prin spații.

**a)** Scrieți un program **C/C++** care, utilizând un algoritm eficient din punct de vedere al timpului de executare și al memoriei utilizate, afișează pe ecran acel termen din șir care îl precede pe cel citit din fișier.

**Exemplu:** dacă fișierul conține valorile alăturate, se va afișa pe ecran numărul 323. **(6p.)**

2 5  
32332

**b)** Descrieți în limbaj natural metoda utilizată și explicați în ce constă eficiența ei. **(4p.)**