# Subiectul III — Exercițiul 1

Informatica — Mate-Info

*Toate aparițiile acestui exercițiu din toate modelele de examen din categoria de mai sus.*

106 exerciții · din 106 sesiuni (2012–2026)

## 1. Informatica — Subiectul III, exercițiul 1

**Sursă: 2012 · Model Oficial**

*2012_Informatica_Mate-Info_Model_Oficial_C-Cpp_Subiect.pdf*

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

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

**1.** Se consideră subprogramul recursiv **CM**, definit alăturat. Pentru a returna valoarea celui mai mare divizor comun al numerelor naturale **100** şi **330**, apelul subprogramului poate fi: **(4p.)**

```
int CM (int x, int y, int z)
{ if(x%z==0 && y%z==0) return z;
  else return CM(x,y,z-1);
}
```

**a.** `CM(1,100,330)`
**b.** `CM(100,330,1)`
**c.** `CM(100,330,100)`
**d.** `CM(330,330,100)`

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

**2.** Având la dispoziție cinci fructe diferite, un **măr**, o **pară**, o **gutuie**, o **caisă**, o **piersică**, se utilizează metoda backtracking pentru a obține toate posibilitățile de a forma grupe de câte trei fructe, ştiind că în cadrul unei grupe nu contează ordinea de aşezare a fructelor. Primele patru soluții obținute sunt, în această ordine: (**măr**, **pară**, **gutuie**), (**măr**, **pară**, **caisă**), (**măr**, **pară**, **piersică**), (**măr**, **gutuie**, **caisă**). Scrieți ultimele două soluții generate, în ordinea obținerii lor. **(6p.)**

**3.** Subprogramul **inserare** are doi parametri:
- **n**, prin care primeşte un număr natural (**2≤n≤20**);
- **a**, prin care primeşte un tablou unidimensional care memorează un şir de **n** numere naturale, fiecare cu cel mult **4** cifre. Cel puțin un element al tabloului este număr par.

Subprogramul modifică tabloul, inserând înainte de fiecare termen par al şirului numărul obținut prin împărțirea la **2** a valorii acestuia şi furnizează, tot prin parametrii **n** şi **a**, valorile actualizate ale datelor primite.
Scrieți în limbajul **C/C++** definiția completă a subprogramului.
**Exemplu:** dacă **n=7** şi **a=(1,4,5,3,82,6,2)** atunci după apel **n=11** şi **a=(1,2,4,5,3,41,82,3,6,1,2)**. **(10p.)**

**4.** Numim număr de tip **palindrom** un număr care are aceeaşi valoare dacă este citit de la stânga la dreapta sau de la dreapta la stânga.
Numim număr de tip **dublu palindrom** o valoare de tip palindrom cu număr par de cifre în care fiecare cifră de rang impar este egală cu cifra alăturată din dreapta ei.
**Exemplu:** **11111111**, **22777722** sunt numere de tip dublu palindrom; **12122121** este număr de tip palindrom, dar nu şi dublu palindrom.
Se cere scrierea în fişierul **BAC.TXT**, fiecare pe câte o linie, a tuturor numerelor naturale de tip dublu palindrom, cu exact **8** cifre. Numerele de tip dublu palindrom sunt scrise în ordine strict crescătoare, iar pentru determinarea acestora se utilizează un algoritm eficient din punct de vedere al timpului de executare.
**a)** Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia. **(4p.)**
**b)** Scrieți programul **C/C++** corespunzător algoritmului descris. **(6p.)**

## 2. Informatica — Subiectul III, exercițiul 1

**Sursă: 2012 · Speciala Subiect**

*2012_Informatica_Mate-Info_Sesiune_Speciala_Subiect.pdf*

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

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

**1.** Algoritmul de generare a tuturor numerelor naturale de `5` cifre, fiecare număr având toate cifrele nenule şi distincte, este echivalent cu algoritmul de generare a: **(4p.)**

**a.** aranjamentelor de `9` elemente luate câte `5`
**b.** combinărilor de `9` elemente luate câte `5`
**c.** permutărilor a `5` elemente
**d.** submulțimilor unei mulțimi cu `9` elemente

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

**2.** Se consideră subprogramul `f`, definit alăturat. Scrieți valorile `f(-6)` şi `f(20)`. **(6p.)**

```
int f(int x)
{ if(x<=0)
    return 0;
  return x+f(x-5);
}
```

**3.** Se consideră subprogramul `minus`, cu doi parametri:
- `n`, prin care primeşte o valoare naturală `2<n<50`;
- `v`, prin care primeşte un tablou unidimensional cu `n` elemente, numere întregi cu cel mult `4` cifre. Cel puțin unul dintre elementele tabloului este nenul.

După fiecare element nenul din tablou, subprogramul inserează câte un nou element, cu aceeaşi valoare absolută, dar cu semn opus, ca în exemplu. Tabloul modificat, precum şi valoarea actualizată a lui `n`, sunt furnizate tot prin parametrii `v`, respectiv `n`.
Scrieți în limbajul `C/C++` definiția completă a subprogramului.
**Exemplu:** dacă `n=5` şi `v=(4, -5, 0, 9, 0)`,
atunci după apel `n=8`, iar `v=(4, -4, -5, 5, 0, 9, -9, 0)`. **(10p.)**

**4.** Fişierul `bac.txt` conține pe prima linie un număr natural **par** `n` cu cel mult patru cifre, iar pe următoarea linie un şir de `n` numere naturale cu cel mult nouă cifre. Numerele din şir sunt în ordine crescătoare şi sunt separate prin câte un spațiu.
Se cere să se afişeze pe ecran cel mai mare număr din prima jumătate a şirului care să fie strict mai mic decât oricare număr din a doua jumătate a şirului. Dacă în fişier nu se află o astfel de valoare, pe ecran se afişează mesajul `Nu exista`.
Pentru determinarea numărului cerut se utilizează un algoritm eficient din punctul de vedere al memoriei şi al timpului de executare.
**Exemplu:** dacă fişierul `bac.txt` are conținutul

```
30
1 3 5 5 ... 5 5 7 10
    ••••••••••
```

de `26` de ori
atunci pe ecran se afişează `3`, iar dacă fişierul are conținutul

```
6
3 3 3 3 9 15
```

atunci pe ecran se afişează `Nu exista`.
**a)** Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia. **(4p.)**
**b)** Scrieți programul `C/C++` corespunzător algoritmului descris. **(6p.)**

## 3. Informatica — Subiectul III, exercițiul 1

**Sursă: 2012 · Toamna August · V04**

*2012_Informatica_Mate-Info_Sesiune_Toamna_August-Septembrie_C-Cpp_Subiect_V04.pdf*

**SUBIECTUL al III-lea** **(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 şirurile de maximum `4` litere distincte din mulțimea {`a`, `v`, `i`, `o`, `n`}. Primele `5` şiruri generate sunt, în această ordine: `a`, `ai`, `ain`, `aino`, `ainv`. Imediat după şirul `avn` se generează: **(4p.)**

**a.** `avi` **b.** `avni` **c.** `avo` **d.** `iano`

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

**2.** Se consideră subprogramul `f`, definit alăturat. Scrieți valorile `f(3,4)` şi `f(2,20)`. **(6p.)**

```
int f(int x,int y)
{ if(x*2>y)
    return x;
  return f(x*2,y);
}
```

**3.** Se consideră subprogramul `zero`, cu doi parametri:

- `n`, prin care primeşte o valoare naturală `2<n<50`;
- `v`, prin care primeşte un tablou unidimensional cu `2•n` elemente, numere naturale cu cel mult `4` cifre. Numărul de elemente pare este egal cu numărul de elemente impare. Elementele au indici de la `1` la `2•n`.

Subprogramul modifică tabloul astfel încât elementele impare să aibă indici impari, iar elementele pare să aibă indici pari. Tabloul modificat este furnizat tot prin parametrul `v`.
Scrieți definiția completă a subprogramului.
**Exemplu:** dacă `n=5` şi `v=(4, 5, 0, 9, 10, 7, 15, 3, 8, 10)`,
atunci după apel o soluție posibilă este: `v=(5, 4, 9, 0, 15, 10, 7, 8, 3, 10)`. **(10p.)**

**4.** Fişierul `bac.txt` conține pe prima linie un număr natural **par** `n` cu cel mult `3` cifre, iar pe următoarea linie un şir de `n` numere naturale cu cel mult nouă cifre. Numerele din şir sunt separate prin câte un spațiu.
Se cere să se afişeze pe ecran mesajul `Da` dacă în şir există un element care să fie strict mai mare decât jumătate dintre numerele din şir.
Dacă în şir nu se află o astfel de valoare, pe ecran se afişează mesajul `Nu`.
Pentru determinarea numărului cerut se utilizează un algoritm eficient din punctul de vedere al memoriei şi al timpului de executare.
**Exemplu:** dacă fişierul `bac.txt` are conținutul

```
6
8 34 34 34 5 34
```

atunci pe ecran se afişează `Nu`, iar dacă fişierul are conținutul

```
8
1 5 6 12 3 12 12 9
```

atunci pe ecran se afişează `Da`.
**a)** Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia. **(4p.)**
**b)** Scrieți programul `C/C++` corespunzător algoritmului descris. **(6p.)**

## 4. Informatica — Subiectul III, exercițiul 1

**Sursă: 2012 · Vara Iunie · V01**

*2012_Informatica_Mate-Info_Sesiune_Vara_Iunie-Iulie_C-Cpp_Subiect_V01.pdf*

**SUBIECTUL al III-lea** **(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 şirurile de `5` cifre din mulțimea {`0`, `1`, `2`}, cu proprietatea că nu există două cifre pare pe poziții consecutive. Primele opt şiruri generate sunt, în această ordine: `01010`, `01011`, `01012`, `01101`, `01110`, `01111`, `01112`, `01121`. Al nouălea şir generat este: **(4p.)**

**a.** `01120` **b.** `01201` **c.** `01210` **d.** `10101`

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

**2.** Se consideră subprogramul `f`, definit alăturat. Scrieți ce se afişează în urma apelului de mai jos.
`f(9);` **(6p.)**

```
void f(int x)
{ cout<<x;   |  printf("%d",x);
  if(x>0)
  { if(x%2==0)
       cout<<'*';   |  printf("*");
    f(x/2);
  }
 }
```

**3.** Se consideră subprogramul `dublu`, cu doi parametri:
- `n`, prin care primeşte o valoare naturală `2<n<50`;
- `v`, prin care primeşte un tablou unidimensional cu `n` elemente, numere întregi cu cel mult `4` cifre. Cel puțin unul dintre elementele tabloului este strict pozitiv.

După fiecare element strict pozitiv din tablou, subprogramul inserează câte un nou element, cu aceeaşi valoare, ca în exemplu. Tabloul modificat, precum şi valoarea actualizată a lui `n`, sunt furnizate tot prin parametrii `v`, respectiv `n`.
Scrieți definiția completă a subprogramului.
**Exemplu:** dacă `n=6` şi `v=(4, -5, 0, 9, 9, -2)`, atunci după apel `n=9`, iar `v=(4, 4, -5, 0, 9, 9, 9, 9, -2)`. **(10p.)**

**4.** Un număr natural cu cel puțin două cifre se numeşte **x-ordonat** dacă toate cifrele sale sunt în ordine crescătoare şi valoarea absolută a diferenței dintre oricare două cifre aflate pe poziții consecutive este egală cu `x`.
**Exemple:** numărul `2468` este 2-ordonat, numărul `147` este 3-ordonat; numerele `179` sau `131` nu sunt de tipul menționat.
Se citeşte de la tastatură un număr natural `x` (`1≤x≤8`) şi se cere scrierea în fişierul `BAC.TXT` a tuturor numerelor naturale distincte **x-ordonate**. Fiecare număr este scris pe câte o linie a fişierului.
Pentru determinarea numerelor cerute se utilizează un algoritm eficient din punctul de vedere al timpului de executare.
**a)** Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia. **(4p.)**
**b)** Scrieți programul `C/C++` corespunzător algoritmului descris. **(6p.)**

## 5. Informatica — Subiectul III, exercițiul 1

**Sursă: 2013 · Model Oficial**

*2013_Informatica_Mate-Info_Model_Oficial_C-Cpp_Subiect.pdf*

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

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

**1.** Se consideră subprogramele **f1** şi **f2**, definite mai jos, în care **a** este nenul.

```
int f1(int a, int b)
{
  return (b/a)*a;
}
```

```
int f2 (int a, int b)
{ if(b%a==0)
    return b;
  return f2(a, b-1);
}
```

Identificați subprogramul care, la apel, pentru parametrii **a=3** şi **b=10**, returnează cel mai mare multiplu al lui **a** mai mic sau egal cu **b**. **(4p.)**

**a.** atât **f1**, cât şi **f2** **b.** numai **f1** **c.** numai **f2** **d.** nici **f1**, nici **f2**

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

**2.** Utilizând metoda bactracking se generează toate numerele cu cifre distincte şi nenule, numere care au suma cifrelor egală cu **10**. Primele patru soluții generate sunt, în această ordine: **1234**, **1243**, **127**, **1324**. Scrieți cea de a cincea şi cea de a şasea soluție, în ordinea generării acestora. **(6p.)**

**3.** Se consideră subprogramul **num**, cu doi parametri:

- **n**, prin care primeşte o valoare naturală **2<n<50**;
- **v**, prin care primeşte un tablou unidimensional cu **n** elemente, numere întregi cu cel mult **4** cifre.

Subprogramul înlocuieşte cu **0** fiecare valoare mai mică sau egală cu prima valoare din tablou. Tabloul modificat este furnizat tot prin parametrul **v**.
Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **n=7** şi **v=(4, 5, 0, 9, 3, 4, -2)**, atunci după apel
**v=(0, 5, 0, 9, 0, 0, 0)**. **(10p.)**

**4.** Fişierul **numere.in** memorează cel mult **1000000** de numere naturale cu cel mult nouă cifre. Numerele sunt ordonate strict crescător şi separate prin câte un spațiu.
Se consideră şirul **1, 4, 7 ....** definit astfel: $f_1=1$, $f_2=4$ şi $f_n=2\cdot f_{n-1}-f_{n-2}$, pentru **n>2**.
Se cere să se afişeze pe ecran numerele din fişier care sunt termeni ai şirului. Numerele sunt afişate în ordine strict crescătoare, separate prin câte un spațiu. Dacă nu există astfel de numere se afişează pe ecran mesajul **Nu exista**.
Pentru determinarea numerelor cerute se utilizează un algoritm eficient din punctul de vedere al memoriei şi al timpului de executare.
**Exemplu:** dacă fişierul **numere.in** conține numerele **1 2 5 7 17 30**, se afişează pe ecran numerele **1 7**.
**a)** Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia. **(4p.)**
**b)** Scrieți programul **C/C++** corespunzător algoritmului descris. **(6p.)**

## 6. Informatica — Subiectul III, exercițiul 1

**Sursă: 2013 · Speciala Subiect**

*2013_Informatica_Mate-Info_Sesiune_Speciala_Subiect.pdf*

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

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

**1.** Se consideră subprogramele **f1** şi **f2**, definite mai jos.

```
int f1(int n)
{ return n*(n+1)/2;
}
```

```
int f2 (int n)
{ if(n>0)
    return n+f2(n-1);
  return 0;
}
```

Identificaţi subprogramul care, la apel, pentru parametrul **n=10**, returnează suma primelor **10** numere naturale. **(4p.)**

**a.** atât **f1**, cât şi **f2** **b.** numai **f1** **c.** numai **f2** **d.** nici **f1**, nici **f2**

**Scrieţi pe foaia de examen răspunsul pentru fiecare dintre cerinţele următoare.**

**2.** Utilizând metoda backtracking se generează toate submulţimile cu cel mult patru elemente din mulţimea {**3**, **5**, **6**, **7**, **8**}. Primele şase soluţii generate sunt, în această ordine: {**3**}, {**3**, **5**}, {**3**, **5**, **6**}, { **3**, **5**, **6**, **7**}, {**3**, **5**, **6**, **8**}, {**3**, **5**, **7**}. Scrieţi cea de a şaptea şi cea de a opta soluţie, în ordinea generării acestora. **(6p.)**

**3.** Se consideră subprogramul **sub**, cu trei parametri:

- **n**, prin care primeşte un număr natural (**2<n<50**);
- **v**, prin care primeşte un tablou unidimensional cu **n** elemente, numere naturale cu cel mult **4** cifre;
- **x**, prin care primeşte un număr natural cu cel mult **4** cifre. Cel puţin unul dintre elementele tabloului are valoarea **x**.

Subprogramul modifică ordinea valorilor din tablou, astfel încât toate valorile egale cu **x** să ocupe primele poziţii din **v**, iar celelalte valori să se regăsească în continuarea acestora. Tabloul modificat este furnizat tot prin parametrul **v**.
Scrieţi definiţia completă a subprogramului.
**Exemplu:** dacă **n=9**, **v=(2**, **1**, **0**, **1**, **7**, **0**, **1**, **4**, **5)** şi **x=1**, atunci, după apel, o soluţie posibilă este **v=(1**, **1**, **1**, **2**, **0**, **7**, **0**, **4**, **5)**. **(10p.)**

**4.** Fişierul **bac.txt** conţine un şir de cel puţin trei şi cel mult **1000000** de numere naturale cu cel mult nouă cifre. Numerele din şir sunt separate prin câte un spaţiu.
Se cere să se afişeze pe ecran, separate printr-un spaţiu, două numere distincte, anume cel mai mic număr par cu două cifre și cel mai mare număr par cu două cifre care **NU** fac parte din şir.
Dacă nu există două astfel de valori, pe ecran se afişează mesajul **Nu exista**.
Pentru determinarea numerelor cerute se utilizează un algoritm eficient din punctul de vedere al timpului de executare.
**Exemplu:** dacă fişierul **bac.txt** conţine valorile
**7 2 40 5 10 15 11 12 18 350**
se afişează pe ecran numerele **14 98**.
**a)** Descrieţi în limbaj natural algoritmul utilizat, justificând eficienţa acestuia. **(4p.)**
**b)** Scrieţi programul **C/C++** corespunzător algoritmului descris. **(6p.)**

## 7. Informatica — Subiectul III, exercițiul 1

**Sursă: 2013 · Toamna August · V06**

*2013_Informatica_Mate-Info_Sesiune_Toamna_August-Septembrie_C-Cpp_Subiect_V06.pdf*

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

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

**1.** Se consideră subprogramul `f`, definit alăturat. Indicaţi ce se afişează în urma apelului de mai jos. **(4p.)**

```
f(3);
```

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

**a.** `12` **b.** `123` **c.** `321` **d.** `3210`

**Scrieţi pe foaia de examen răspunsul pentru fiecare dintre cerinţele următoare.**

**2.** Utilizând metoda backtracking, se generează în ordine lexicografică toate şirurile de câte `5` litere distincte din mulţimea {`A`, `B`, `C`, `D`, `E`}, astfel încât în fiecare şir litera `D` precede literele `A` şi `B`. Primele cinci soluţii generate sunt, în această ordine: `CDABE`, `CDAEB`, `CDBAE`, `CDBEA`, `CDEAB`. Scrieţi cea de a şasea şi cea de a şaptea soluţie, în ordinea generării acestora. **(6p.)**

**3.** Se consideră subprogramul `sub`, cu trei parametri:
- `n`, prin care primeşte un număr natural (`2<n<50`);
- `v`, prin care primeşte un tablou unidimensional cu `n` elemente, numere naturale cu cel mult `4` cifre;
- `k`, prin care primeşte un număr natural (`1<k≤n`).

Subprogramul returnează suma primelor `k` elemente cu valoare impară ale tabloului. Dacă nu există `k` elemente impare în tablou, subprogramul returnează valoarea `-1`.
Scrieţi definiţia completă a subprogramului.
**Exemplu:** dacă `n=8`, `v=(2, 7, 6, 8, 3, 7, 5, 1)`, `k=3`, atunci subprogramul returnează valoarea `17` (`7+3+7=17`). **(10p.)**

**4.** Fiind date două numere `a` şi `b`, îl numim pe `a` sufix al lui `b` dacă `a` este egal cu `b` sau dacă `b` se poate obţine din `a` prin alipirea la stânga a unor noi cifre.
**Exemplu:** `12` este sufix al lui `12`, iar `15` este sufix al lui `31415`.
Fişierul `bac.txt` conţine pe prima linie un număr natural `x`, cu cel mult nouă cifre, iar pe a doua linie un şir de cel puţin două şi cel mult `1000000` de numere naturale cu cel mult nouă cifre. Numerele din şir sunt separate prin câte un spaţiu.
Se cere să se afişeze pe ecran ultimul termen al şirului care are ca sufix numărul `x`. Dacă în şir nu există o astfel de valoare, pe ecran se afişează mesajul `Nu exista`.
Pentru determinarea numărului cerut se utilizează un algoritm eficient din punctul de vedere al memoriei şi al timpului de executare.
**Exemplu:** dacă fişierul `bac.txt` conţine numerele

```
12
3445 89312 1245 12 67120 312 1234578
```

atunci pe ecran se afişează `312`.
**a)** Descrieţi în limbaj natural algoritmul utilizat, justificând eficienţa acestuia. **(4p.)**
**b)** Scrieţi programul `C/C++` corespunzător algoritmului descris. **(6p.)**

## 8. Informatica — Subiectul III, exercițiul 1

**Sursă: 2013 · Vara Iunie · V02**

*2013_Informatica_Mate-Info_Sesiune_Vara_Iunie-Iulie_C-Cpp_Subiect_V02.pdf*

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

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

**1.** Se consideră subprogramele **f1** şi **f2**, definite mai jos.

```
int f1 (int x, int y)
{
   return x*y;
}
```

```
int f2 (int x, int y)
{ if (y==1) return x;
  else return x*f2(x, y-1);
}
```

La apel, pentru parametrii **x=2** şi **y=3**, returnează $x^y$: **(4p.)**

**a.** atât **f1**, cât şi **f2** **b.** numai **f1** **c.** numai **f2** **d.** nici **f1**, nici **f2**

**Scrieţi pe foaia de examen răspunsul pentru fiecare dintre cerinţele următoare.**

**2.** Utilizând metoda backtracking se generează toate cuvintele de **5** litere din mulţimea {**A**, **M**}, cuvinte care încep şi se termină cu **M**. Primele patru soluţii generate sunt, în această ordine: **MAAAM**, **MAAMM**, **MAMAM**, **MAMMM**. Scrieţi cea de a cincea şi cea de a şasea soluţie, în ordinea generării acestora. **(6p.)**

**3.** Se consideră subprogramul **valuri**, cu doi parametri:

- **n**, prin care primeşte o valoare naturală **2<n<50**;
- **v**, prin care furnizează un tablou unidimensional cu **2·n** elemente, valori întregi din intervalul **[1, 2·n]**.

Subprogramul construieşte tabloul **v** astfel încât, în acesta, şirul elementelor impare să fie strict crescător, iar şirul elementelor pare să fie strict descrescător. Primul element al tabloului este impar, iar două elemente cu aceeaşi paritate nu pot ocupa poziţii consecutive în tablou, ca în exemplu.
Scrieţi definiţia completă a subprogramului.
**Exemplu:** dacă **n=4** atunci, după apel, **v=(1, 8, 3, 6, 5, 4, 7, 2)**. **(10p.)**

**4.** Se consideră şirul definit alăturat (unde **n** este un număr natural nenul), în care nu există doi termeni cu aceeași paritate aflați pe poziții consecutive:
**1**, **2**, **3**, **4**, **7**, **8**, **15**, **16** ....

$$f_n = \begin{cases} 1, & \text{dacă } n = 1 \\ 1 + f_{n-1}, & \text{dacă } n \text{ par} \\ 1 + 2 \cdot f_{n-2}, & \text{altfel} \end{cases}$$

Se citeşte de la tastatură un număr natural **x**, cu cel mult nouă cifre, termen al şirului dat, şi se cere să se scrie în fişierul text **bac.txt**, în ordine strict descrescătoare, separați prin câte un spațiu, toţi termenii şirului care sunt mai mici sau egali cu **x**.
Pentru determinarea termenilor ceruţi se utilizează un algoritm eficient din punctul de vedere al memoriei utilizate şi al timpului de executare.
**Exemplu:** dacă **x=15**, fişierul **bac.txt** conţine numerele
**15 8 7 4 3 2 1**
**a)** Descrieţi în limbaj natural algoritmul utilizat, justificând eficienţa acestuia. **(4p.)**
**b)** Scrieţi programul **C/C++** corespunzător algoritmului descris. **(6p.)**

## 9. Informatica — Subiectul III, exercițiul 1

**Sursă: 2014 · Model Oficial**

*2014_Informatica_Mate-Info_Model_Oficial_C-Cpp_Subiect.pdf*

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

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

**1.** Se consideră subprogramele **f1** şi **f2**, definite mai jos.

```
long f1 (int n)
{ if(n<10)
    return n*f1(n+1);
  return 10;
}
```

```
long f2 (int n)
{ if(n>1)
    return n*f2(n-1);
  return 1;
}
```

Identificaţi subprogramul care, la apel, pentru parametrul **n=10**, returnează produsul primelor **10** numere naturale nenule. **(4p.)**

**a.** atât **f1**, cât şi **f2** **b.** numai **f1** **c.** numai **f2** **d.** nici **f1**, nici **f2**

**Scrieţi pe foaia de examen răspunsul pentru fiecare dintre cerinţele următoare.**

**2.** Alfabetul Morse este format doar din simbolurile linie (notat cu **−**) şi punct (notat cu **•**). Utilizând metoda bactracking se generează toate cuvintele scrise în alfabetul Morse, formate din câte **5** simboluri, care încep şi se termină cu punct. Primele patru soluţii generate sunt, în această ordine:

• − − − •  • − − • •  • − • − •  • − • • •

Scrieţi cea de a cincea şi cea de a şasea soluţie, în ordinea generării acestora. **(6p.)**

**3.** Se consideră subprogramul **divizor**, cu doi parametri:

- **n**, prin care primește un număr natural cu cel mult nouă cifre (**n>1**);
- **d**, prin care furnizează cel mai mare divizor prim al lui **n**.

Scrieţi definiţia completă a subprogramului.
**Exemplu:** pentru **n=50**, după apel **d=5**, iar pentru **n=11**, după apel **d=11**. **(10p.)**

**4.** Fişierul **bac.txt** conţine pe prima linie un număr natural **par n** cu cel mult patru cifre, iar pe următoarea linie un şir de **n** numere naturale cu cel mult nouă cifre. Numerele din şir sunt în ordine descrescătoare şi sunt separate prin câte un spaţiu.
Se cere să se afişeze pe ecran cel mai mic număr din prima jumătate a şirului care să fie strict mai mare decât oricare număr din a doua jumătate a şirului. Dacă în fişier nu se află o astfel de valoare, pe ecran se afişează mesajul **Nu exista**.
Pentru determinarea numărului cerut se utilizează un algoritm eficient din punctul de vedere al memoriei şi al timpului de executare.
**Exemplu:** dacă fişierul **bac.txt** are conţinutul

```
30
16 8 7 7 ... 7 7 2 1
```

(**7 7 ... 7 7** de **26** de ori)

atunci pe ecran se afişează **8**, iar dacă fişierul are conţinutul

```
6
7 7 7 7 2 1
```

atunci pe ecran se afişează **Nu exista**.
**a)** Descrieţi în limbaj natural algoritmul utilizat, justificând eficienţa acestuia. **(4p.)**
**b)** Scrieţi programul **C/C++** corespunzător algoritmului descris. **(6p.)**

## 10. Informatica — Subiectul III, exercițiul 1

**Sursă: 2014 · Simulare Nationala**

*2014_Informatica_Mate-Info_Simulare_Nationala_C-Cpp_Subiect.pdf*

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

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

**1.** Utilizând metoda backtracking se generează, în ordine crescătoare, toate numerele de patru cifre din mulțimea {**1**, **4**, **6**, **8**, **9**}, cu proprietatea că nu au două cifre alăturate egale. Primele patru numere generate sunt, în această ordine: **1414**, **1416**, **1418**, **1419**. Al cincilea număr generat este: **(4p.)**

**a.** **1441** **b.** **1461** **c.** **1468** **d.** **1491**

**Scrieţi pe foaia de examen răspunsul pentru fiecare dintre cerinţele următoare.**

**2.** Se consideră subprogramul **f**, definit alăturat. Scrieţi ce se afişează în urma apelului de mai jos.
**f(6);**
**(6p.)**

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

**3.** Se consideră subprogramul **divizori**, cu trei parametri:

- **n**, prin care primeşte o valoare naturală ($0<n<10^6$);
- **mic** şi **mare**, prin care furnizează cel mai mic şi respectiv cel mai mare divizor propriu pozitiv al lui **n** (diferit de **1** şi de **n**) sau valoarea **0**, prin amândoi, dacă numărul nu are cel puţin doi astfel de divizori.

Scrieţi definiţia completă a subprogramului.
**Exemplu**: pentru numărul **n=12**, în urma apelului, **mic=2** şi **mare=6**, iar pentru numărul **n=9**, în urma apelului, **mic=0** şi **mare=0**. **(10p.)**

**4.** Se consideră un șir ai cărui termeni sunt numere naturale nenule, de o singură cifră. Numim **număr asociat** al acestui șir un număr natural format cu termenii șirului, în ordinea în care aceștia apar în șir.
**Exemplu:** numărul asociat șirului **1**, **2**, **5**, **3**, **2** este **12532**.
Fişierul text **bac.txt** conţine un șir de cel puţin trei şi cel mult **80** de termeni, numere naturale nenule, de o singură cifră, separate prin câte un spaţiu.
Se cere determinarea unui șir obținut prin eliminarea a doi termeni situați pe poziții consecutive în șirul aflat în fișier, astfel încât numărul asociat șirului obținut să fie maxim. Termenii șirului obținut se afișează pe ecran, separați prin câte un spațiu.
Se utilizează un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.
**Exemplu:** dacă fişierul **bac.txt** conţine șirul
**9 8 5 7 6 2 3 4**
atunci, pentru că numerele asociate șirurilor care se pot obține sunt **576234**, **976234**, **986234**, **985234**, **985734**, **985764**, **985762**, pe ecran se afişează șirul:
**9 8 6 2 3 4**

**a)** Descrieţi în limbaj natural algoritmul utilizat, justificând eficienţa acestuia. **(4p.)**
**b)** Scrieţi programul **C/C++** corespunzător algoritmului descris. **(6p.)**

## 11. Informatica — Subiectul III, exercițiul 1

**Sursă: 2014 · Speciala C · V02**

*2014_Informatica_Mate-Info_Sesiune_Speciala_C-Cpp_Subiect_V02.pdf*

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

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

**1.** Utilizând metoda backtracking, se generează toate posibilitățile de a obține suma **4** cu numere naturale nenule. Două sume sunt distincte dacă diferă prin cel puțin un termen. Soluţiile generate sunt, în această ordine, **1+1+1+1**, **1+1+2**, **1+3**, **2+2**.
Aplicând același algoritm pentru a genera toate posibilitățile de a obține suma **6**, dacă prima soluție generată este **1+1+1+1+1+1**, atunci soluția **1+2+3** este generată: **(4p.)**

**a.** a **6**-a **b.** a **7**-a **c.** a **8**-a **d.** a **9**-a

**Scrieţi pe foaia de examen răspunsul pentru fiecare dintre cerinţele următoare.**

**2.** Se consideră subprogramul **f**, definit alăturat. Scrieţi valorile **f(5,5)** şi **f(10,21)**. **(6p.)**

```
int f(int a, int b)
{ if (a==b) return 0;
  if (b/a==0) return a+b;
  return f(a+2,b-3);
}
```

**3.** Un număr natural nenul se numește **perfect** dacă este egal cu suma divizorilor săi naturali strict mai mici decât el.
**Exemplu:** **28** este număr perfect pentru că **28=1+2+4+7+14**.
Se consideră subprogramul **perfect**, cu doi parametri, **a** și **b**, prin care primeşte câte un număr natural (**2≤a<b≤10⁹**). Subprogramul afișează pe ecran, separate prin câte un spațiu, în ordine descrescătoare, toate numerele perfecte din intervalul **[a,b]**. Dacă în interval nu există astfel de numere, subprogramul afișează pe ecran mesajul **nu exista**.
Scrieţi definiţia completă a subprogramului.
**Exemplu:** pentru **a=5** și **b=30**, se afișează pe ecran: **28 6** **(10p.)**

**4.** Numim **secvență uniformă** a unui șir de numere naturale un subșir al acestuia, format din termeni cu aceeași valoare, aflați pe poziții consecutive în șirul dat. Lungimea secvenței este egală cu numărul de termeni ai acesteia.
Fișierul **bac.txt** conține un șir de cel puțin două și cel mult **1000000000** de numere naturale din intervalul [**0**, **10⁹**]. Numerele sunt separate prin câte un spațiu, iar în șir există cel puțin doi termeni egali pe poziții consecutive.
Se cere să se determine o secvență uniformă de lungime maximă în șirul aflat în fișier și să se afișeze pe ecran lungimea acestei secvențe și, pe o linie nouă, separați prin câte un spațiu, termenii acesteia. Dacă sunt mai multe astfel de secvențe, se afișează doar termenii ultimei dintre acestea. Pentru determinarea numerelor cerute se utilizează un algoritm eficient din punctul de vedere al memoriei necesare și al timpului de executare.
**Exemplu:** dacă fişierul **bac.txt** conţine numerele
**2 3 3 3 3 5 4 4 11 11 11 11 16 11 11 11 11 15 15**
atunci pe ecran se afișează valorile
**4**
**11 11 11 11**
**a)** Descrieţi în limbaj natural algoritmul utilizat, justificând eficienţa acestuia. **(4p.)**
**b)** Scrieţi programul **C/C++** corespunzător algoritmului descris. **(6p.)**

## 12. Informatica — Subiectul III, exercițiul 1

**Sursă: 2014 · Toamna August · V10**

*2014_Informatica_Mate-Info_Sesiune_Toamna_August-Septembrie_C-Cpp_Subiect_V10.pdf*

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

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

**1.** Utilizând metoda backtracking, se generează toate posibilitățile de a forma succesiuni de câte `5` genuri muzicale distincte din mulțimea {`jazz`, `rock`, `latino`, `house`, `pop`}, astfel încât în fiecare succesiune genul latino precede genul house. Două succesiuni sunt distincte dacă genurile muzicale sunt în altă ordine.
Primele cinci soluţii generate sunt, în această ordine, (`jazz`, `rock`, `latino`, `house`, `pop`), (`jazz`, `rock`, `latino`, `pop`, `house`), (`jazz`, `rock`, `pop`, `latino`, `house`), (`jazz`, `latino`, `rock`, `house`, `pop`), (`jazz`, `latino`, `rock`, `pop`, `house`). Imediat înainte de (`pop`, `latino`, `house`, `jazz`, `rock`) este generată soluția: **(4p.)**

**a.** (`rock`, `jazz`, `house`, `latino`, `pop`)
**b.** (`rock`, `jazz`, `latino`, `house`, `pop`)
**c.** (`pop`, `latino`, `rock`, `house`, `jazz`)
**d.** (`pop`, `rock`, `latino`, `house`, `jazz`)

**Scrieţi pe foaia de examen răspunsul pentru fiecare dintre cerinţele următoare.**

**2.** Se consideră subprogramul `f`, definit alăturat. Scrieți două valori naturale distincte din intervalul `[1,50]` pe care le poate avea variabila întreagă `x`, astfel încât `f(30,x)` să aibă valoarea `5`. **(6p.)**

```
int f(int a,int b)
{ if (b==0) return a;
  else return f(b,a%b);
}
```

**3.** Se consideră subprogramul `triplete`, cu un singur parametru, `n`, prin care primeşte o valoare naturală din intervalul $[2,10^4]$. Subprogramul afișează pe ecran toate tripletele de numere naturale `(x, y, z)` cu proprietatea că $x<y<z$ și $x \cdot y+y \cdot z=n$. Fiecare triplet se afișează pe câte o linie a ecranului, iar numerele din fiecare triplet sunt separate prin câte o virgulă și încadrate între paranteze rotunde, ca în exemplu.
Scrieţi definiţia completă a subprogramului.
**Exemplu:** pentru `n=8` se afișează pe ecran, nu neapărat în această ordine, tripletele:
`(0,1,8)`
`(0,2,4)`
`(1,2,3)` **(10p.)**

**4.** Fișierul `bac.txt` conține pe prima linie un număr natural `n` ($1 \le n \le 10^6$), iar pe a doua linie cel mult `1000000` de numere naturale de forma $10^p$ ($0 \le p \le 9$), separate prin câte un spațiu. Se cere să se afișeze pe ecran numărul care ar apărea pe poziția `n` în șirul ordonat crescător obținut din toate numerele aflate pe a doua linie a fișierului. Dacă șirul are mai puțin de `n` termeni, se afișează pe ecran mesajul `Nu exista`.
Pentru determinarea numărului cerut se utilizează un algoritm eficient din punctul de vedere al timpului de executare.
**Exemplu:** dacă fişierul `bac.txt` conţine numerele
`5`
`100 100000 1 100000 1000 100 10`
atunci pe ecran se afișează valoarea
`1000`
**a)** Descrieţi în limbaj natural algoritmul utilizat, justificând eficienţa acestuia. **(4p.)**
**b)** Scrieţi programul `C/C++` corespunzător algoritmului descris. **(6p.)**

## 13. Informatica — Subiectul III, exercițiul 1

**Sursă: 2014 · Vara Iunie · V04**

*2014_Informatica_Mate-Info_Sesiune_Vara_Iunie-Iulie_C-Cpp_Subiect_V04.pdf*

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

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

**1.** Se consideră subprogramul **f**, definit alăturat. Indicați ce valoare are **f(15)**. **(4p.)**

```
int f(int n)
{ if (n<10) return f(n+1)+3;
  else if (n==10) return 7;
       else return f(n-2)-1;
}
```

**a.** 1 **b.** 7 **c.** 8 **d.** 10

**Scrieţi pe foaia de examen răspunsul pentru fiecare dintre cerinţele următoare.**

**2.** Utilizând metoda backtracking, se generează toate posibilitățile de a forma șiraguri de câte **4** mărgele de culori distincte din mulţimea {**roșu**, **galben**, **roz**, **albastru**, **violet**}, astfel încât în fiecare șirag nu pot fi pe poziții alăturate mărgele roșii și galbene. Două șiraguri sunt distincte dacă au cel puțin o mărgea de culoare diferită sau dacă ordinea culorilor mărgelelor este diferită.
Primele cinci soluţii generate sunt, în această ordine, (**roșu**, **roz**, **galben**, **albastru**), (**roșu**, **roz**, **galben**, **violet**), (**roșu**, **roz**, **albastru**, **galben**), (**roșu**, **roz**, **albastru**, **violet**), (**roșu**, **roz**, **violet**, **galben**). Scrieţi cea de a şasea şi cea de a şaptea soluţie, în ordinea generării acestora. **(6p.)**

**3.** Un interval cu proprietatea că există un singur număr natural, **n** (**2≤n**), pentru care valoarea produsului **1·2·3·…·n** aparține acestui interval este numit **interval factorial** al lui **n**.
**Exemplu:** **[5,8]** și **[3,23]** sunt intervale factoriale ale lui **3**, dar **[1,15]** și **[7,10]** nu sunt intervale factoriale ale niciunui număr.
Se consideră subprogramul **interval**, cu trei parametri:

- **n**, prin care primește un număr natural din intervalul **[2,10]**.
- **a** și **b**, prin care furnizează câte un număr natural, astfel încât expresia **b-a** să aibă valoare maximă, iar **[a,b]** să fie interval factorial al lui **n**.

Scrieţi definiţia completă a subprogramului.
**Exemplu:** dacă **n=3**, după apel **a=3** și **b=23**. **(10p.)**

**4.** Un număr natural **x**, format din exact două cifre, este numit **sub-număr** al unui număr natural **y** dacă cifrele lui **x** apar, în aceeași ordine, pe ranguri consecutive, în numărul **y**.
**Exemplu:** **21** este sub-număr al lui **12145**, al lui **213**, al lui **21**, dar nu și al lui **123** sau al lui **231**.
Fișierul **bac.txt** conține cel mult **1000000** de numere naturale din intervalul [**10**, $\mathbf{10^9}$], separate prin câte un spațiu.
Se cere să se afișeze pe ecran, separate prin câte un spațiu, sub-numerele care apar de cele mai multe ori în scrierea numerelor din fișier. Pentru determinarea sub-numerelor cerute se utilizează un algoritm eficient din punctul de vedere al timpului de executare.
**Exemplu:** dacă fişierul **bac.txt** conţine numerele
**393 17775787 72194942 12121774**
atunci pe ecran se afișează valorile de mai jos, nu neapărat în această ordine:
**77 21**
**a)** Descrieţi în limbaj natural algoritmul utilizat, justificând eficienţa acestuia. **(4p.)**
**b)** Scrieţi programul **C/C++** corespunzător algoritmului descris. **(6p.)**

## 14. Informatica — Subiectul III, exercițiul 1

**Sursă: 2015 · Model Oficial**

*2015_Informatica_Mate-Info_Model_Oficial_C-Cpp_Subiect.pdf*

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

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

**1.** Utilizând metoda backtracking, se generează toate posibilitățile de a forma succesiuni de câte `5` tipuri de creații literare din mulțimea {**baladă**, **basm**, **eseu**, **epopee**, **poem**}, astfel încât în fiecare succesiune eseul precede epopeea. Două succesiuni sunt distincte dacă tipurile de creații literare sunt în altă ordine.
Primele cinci soluţii generate sunt, în această ordine, (**baladă**, **basm**, **eseu**, **epopee**, **poem**), (**baladă**, **basm**, **eseu**, **poem**, **epopee**), (**baladă**, **basm**, **poem**, **eseu**, **epopee**), (**baladă**, **eseu**, **basm**, **epopee**, **poem**), (**baladă**, **eseu**, **basm**, **poem**, **epopee**). Imediat înainte de (**poem**, **eseu**, **epopee**, **baladă**, **basm**) este generată soluția: **(4p.)**

**a.** (**basm**, **baladă**, **epopee**, **eseu**, **poem**) **b.** (**basm**, **baladă**, **eseu**, **epopee**, **poem**)

**c.** (**poem**, **eseu**, **basm**, **epopee**, **baladă**) **d.** (**poem**, **basm**, **eseu**, **epopee**, **baladă**)

**Scrieţi pe foaia de examen răspunsul pentru fiecare dintre cerinţele următoare.**

**2.** Se consideră subprogramul `f`, definit alăturat. Scrieţi valorile `f(10,5)` şi `f(20,9)`. **(6p.)**

```
int f(int a, int b)
{ if (a<b) return a;
  if (a%b==0) return b;
  return f(a-1,b+2);
}
```

**3.** Un număr natural nenul se numește **subperfect** dacă este strict mai mic decât suma divizorilor săi proprii (divizori naturali diferiți de `1` și de el însuși).
**Exemplu:** `12` este număr subperfect pentru că `12<2+3+4+6`.
Se consideră subprogramul **`subperfect`**, cu doi parametri, `a` și `b`, prin care primeşte câte un număr natural ($2 \le a < b \le 10^9$). Subprogramul afișează pe ecran, separate prin câte un spațiu, în ordine descrescătoare, toate numerele subperfecte din intervalul `[a,b]`. Dacă în interval nu există astfel de numere, subprogramul afișează pe ecran mesajul **`nu exista`**.
Scrieţi definiţia completă a subprogramului.
**Exemplu:** pentru `a=10` și `b=20`, se afișează pe ecran: `20 18 12` **(10p.)**

**4.** Fișierul **`bac.txt`** conține pe prima linie un număr natural, `n` ($1 \le n \le 10^6$), iar pe a doua linie cel mult `1000000` de numere naturale de forma $2^p$ ($0 \le p \le 9$), separate prin câte un spațiu.
Se cere să se afișeze pe ecran numărul care ar apărea pe poziția `n` în șirul ordonat descrescător obținut din toate numerele aflate pe a doua linie a fișierului. Dacă șirul are mai puțin de `n` termeni, se afișează pe ecran mesajul **`Nu exista`**.
Pentru determinarea numărului cerut se utilizează un algoritm eficient din punctul de vedere al timpului de executare.
**Exemplu:** dacă fişierul **`bac.txt`** conţine numerele
`5`
`16 32 1 64 128 32 128 32`
atunci pe ecran se afișează valoarea
`32`
**a)** Descrieţi în limbaj natural algoritmul utilizat, justificând eficienţa acestuia. **(4p.)**
**b)** Scrieţi programul `C/C++` corespunzător algoritmului descris. **(6p.)**

## 15. Informatica — Subiectul III, exercițiul 1

**Sursă: 2015 · Simulare Nationala**

*2015_Informatica_Mate-Info_Simulare_Nationala_C-Cpp_Subiect.pdf*

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

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

**1.** Utilizând metoda backtracking, se generează toate aranjamentele florale de câte **5** flori distincte din mulţimea {**frezie**, **iris**, **lalea**, **mac**, **nard**}. Două aranjamente sunt distincte dacă florile sunt dispuse în altă ordine.
Primele patru soluţii obţinute sunt, în această ordine: (**frezie**, **iris**, **lalea**, **mac**, **nard**), (**frezie**, **iris**, **lalea**, **nard**, **mac**), (**frezie**, **iris**, **mac**, **lalea**, **nard**), (**frezie**, **iris**, **mac**, **nard**, **lalea**). A cincea soluție generată este: **(4p.)**

**a.** (**frezie**, **iris**, **nard**, **lalea**, **mac**)
**b.** (**frezie**, **iris**, **nard**, **mac**, **lalea**)
**c.** (**frezie**, **lalea**, **iris**, **mac**, **nard**)
**d.** (**frezie**, **lalea**, **iris**, **nard**, **mac**)

**Scrieţi pe foaia de examen răspunsul pentru fiecare dintre cerinţele următoare.**

**2.** Subprogramul **F** este definit alăturat. Scrieţi ce valori au **F(105,105)** respectiv **F(105,42)**. **(6p.)**

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

**3.** Subprogramul **Cifre** are un singur parametru, **n**, prin care primește un număr natural ($n \in [0, 10^9]$). Subprogramul afișează pe ecran toate cifrele distincte care apar în scrierea lui **n**, separate prin câte un spațiu.
Scrieţi definiţia completă a subprogramului.
**Exemplu:** dacă **n=24207**, se afișează cifrele **0 2 4 7**, nu neapărat în această ordine. **(10p.)**

**4.** Fișierul **bac.txt** conține numere naturale din intervalul $[1, 10^4]$: pe prima linie numărul **n**, pe a doua linie un șir de **n** numere ordonate strict descrescător, iar pe a treia linie două numere, **x** și **y** (**x**≤**y**). Numerele de pe aceeași linie sunt separate prin câte un spațiu.
Se cere să se afișeze pe ecran cel mai mare număr din șir care aparține intervalului **[x,y]**. Dacă nu există un astfel de număr, se afișează pe ecran mesajul **nu exista**. Pentru determinarea numărului cerut se utilizează un algoritm eficient din punctul de vedere al timpului de executare.
**Exemplu:** dacă fișierul conține numerele

```
5
100 49 16 7 2
10 30
```

atunci pe ecran se afișează

```
16
```

**a)** Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia. **(4p.)**
**b)** Scrieți programul **C/C++** corespunzător algoritmului descris. **(6p.)**

## 16. Informatica — Subiectul III, exercițiul 1

**Sursă: 2015 · Speciala C · V05**

*2015_Informatica_Mate-Info_Sesiune_Speciala_C-Cpp_Subiect_V05.pdf*

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

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

**1.** Utilizând metoda backtracking, se generează toate șiragurile formate din câte **5** pietre distincte din mulţimea {**opal**, **rubin**, **safir**, **smarald**, **topaz**}. Două șiraguri sunt distincte dacă pietrele sunt dispuse în altă ordine. Primele patru soluţii obţinute sunt, în această ordine: (**opal**, **rubin**, **safir**, **smarald**, **topaz**), (**opal**, **rubin**, **safir**, **topaz**, **smarald**), (**opal**, **rubin**, **smarald**, **safir**, **topaz**) şi (**opal**, **rubin**, **smarald**, **topaz**, **safir**). Indicați soluția care trebuie eliminată din enumerarea următoare, astfel încât cele rămase să apară în ordinea generării lor, pe poziții consecutive: (**smarald**, **safir**, **opal**, **topaz**, **rubin**) (**smarald**, **safir**, **topaz**, **opal**, **rubin**) (**smarald**, **safir**, **topaz**, **rubin**, **opal**) (**smarald**, **topaz**, **opal**, **rubin**, **safir**). **(4p.)**

**a.** (**smarald**, **safir**, **opal**, **topaz**, **rubin**) **b.** (**smarald**, **safir**, **topaz**, **opal**, **rubin**)

**c.** (**smarald**, **safir**, **topaz**, **rubin**, **opal**) **d.** (**smarald**, **topaz**, **opal**, **rubin**, **safir**)

**Scrieţi pe foaia de examen răspunsul pentru fiecare dintre cerinţele următoare.**

**2.** Subprogramul **F** este definit alăturat. Scrieţi ce se afişează în urma apelului de mai jos.

```
F(154678,3);
```

**(6p.)**

```
void F (long a, int b)
{ if(a*b!=0)
    if(a%2==0)
    { cout<<a%10;   |  printf("%d",a%10);
      F(a/10,b-1);
    }
    else
    { F(a/10,b+1);
      cout<<a%10;   |  printf("%d",a%10);
    }
}
```

**3.** Subprogramul **DivImpar** are doi parametri, **a** şi **b**, prin care primeşte două numere naturale din intervalul $[1,10^5]$. Subprogramul returnează cel mai mare divizor comun impar al numerelor **a** şi **b**.
Scrieţi definiţia completă a subprogramului.
**Exemplu:** dacă **a=30** şi **b=60**, subprogramul returnează valoarea **15**. **(10p.)**

**4.** Fişierul **date.in** conţine un şir de cel mult un milion de numere naturale din intervalul $[0,10^9]$, separate prin câte un spațiu. Șirul are cel puțin doi termeni pari și cel puțin doi termeni impari.
Se cere să se afișeze pe ecran mesajul **DA** dacă șirul aflat în fișier are un subșir ordonat crescător, format din toți termenii pari ai săi, și un subșir ordonat descrescător, format din toți termenii impari ai săi. Dacă nu există două astfel de subșiruri, programul afișează pe ecran mesajul **NU**. Pentru verificarea proprietății cerute utilizați un algoritm eficient din punctul de vedere al timpului de executare și al memoriei necesare.
**Exemplu:** dacă fișierul **date.in** conține numerele
**7 2 5 2 4 3 8**
se afișează pe ecran mesajul
**DA**
iar dacă fișierul conține numerele
**5 2 7 2 4 3 8**
se afișează pe ecran mesajul
**NU**
**a)** Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia. **(4p.)**
**b)** Scrieți programul C/C++ corespunzător algoritmului descris. **(6p.)**

## 17. Informatica — Subiectul III, exercițiul 1

**Sursă: 2015 · Toamna August · V02**

*2015_Informatica_Mate-Info_Sesiune_Toamna_August-Septembrie_C-Cpp_Subiect_V02.pdf*

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

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

**1.** Utilizând metoda backtracking, se generează toate parfumurile formate prin amestecarea a câte **3** esențe distincte din mulţimea {**ambră**, **cedru**, **iris**, **mosc**, **santal**}. Primele patru soluţii obţinute sunt, în această ordine: (**ambră**, **cedru**, **iris**), (**ambră**, **cedru**, **mosc**), (**ambră**, **cedru**, **santal**) şi (**ambră**, **iris**, **mosc**). Indicaţi submulţimea care trebuie eliminată din enumerarea următoare, astfel încât soluţiile rămase să apară în ordinea generării lor: (**ambră**, **mosc**, **santal**), (**cedru**, **mosc**, **santal**), (**cedru**, **iris**, **mosc**), (**cedru**, **iris**, **santal**). **(4p.)**

**a.** (**ambră**, **mosc**, **santal**)
**b.** (**cedru**, **mosc**, **santal**)
**c.** (**cedru**, **iris**, **mosc**)
**d.** (**cedru**, **iris**, **santal**)

**Scrieţi pe foaia de examen răspunsul pentru fiecare dintre cerinţele următoare.**

**2.** Subprogramul **F** este definit alăturat. Scrieţi instrucţiunea prin care se poate apela subprogramul pentru a afişa, în ordine strict descrescătoare, toţi divizorii pozitivi proprii ai numărului **2015** (divizori naturali diferiți de **1** și de **2015**). **(6p.)**

```
void F (int n, int d)
{ if(d<n/2) F(n, d+1);
  if(n%d==0)
    cout<<d<<' '; | printf("%d ",d);
}
```

**3.** Subprogramul **NrPrime** are un singur parametru, **n**, prin care primeşte un număr natural ($n \in [0, 10^9]$). Subprogramul returnează numărul de cifre prime ale lui **n**.
Scrieţi definiţia completă a subprogramului.
**Exemplu:** dacă **n=1233405**, atunci subprogramul returnează valoarea **4**. **(10p.)**

**4.** Fişierul **BAC.TXT** conţine pe prima linie un număr natural, **n** ($n \in [2, 5000]$), și pe a doua linie un șir de $2 \cdot n$ numere naturale din intervalul $[0, 5]$. Numerele aflate pe aceeași linie a fișierului sunt separate prin câte un spaţiu.
Se cere să se afişeze pe ecran valoarea obținută însumând **toate** produsele de forma $x \cdot y$, unde **x** și **y** sunt numere de paritate diferită, **x** fiind printre primii **n** termeni ai șirului aflat în fișier, iar **y** printre ultimii **n** termeni ai acestui șir. Dacă nu există niciun astfel de produs, valoarea cerută este nulă. Pentru determinarea numărului cerut utilizați un algoritm eficient din punctul de vedere al timpului de executare și al memoriei necesare.
**Exemplu:** dacă fişierul are conţinutul de mai jos
**5**
**1 2 0 0 5 4 2 3 1 0**
pe ecran se afişează numărul **44**
($1 \cdot 4+1 \cdot 2+1 \cdot 0+2 \cdot 3+2 \cdot 1+0 \cdot 3+0 \cdot 1+0 \cdot 3+0 \cdot 1+5 \cdot 4+5 \cdot 2+5 \cdot 0=44$).
**a)** Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia. **(4p.)**
**b)** Scrieți programul C/C++ corespunzător algoritmului descris. **(6p.)**

## 18. Informatica — Subiectul III, exercițiul 1

**Sursă: 2015 · Vara Iunie · V09**

*2015_Informatica_Mate-Info_Sesiune_Vara_Iunie-Iulie_C-Cpp_Subiect_V09.pdf*

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

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

**1.** Utilizând metoda backtracking, se generează toate numerele naturale din intervalul `[100,999]` care au suma cifrelor egală cu `5`. Primele cinci soluţii obţinute sunt, în această ordine `104`, `113`, `122`, `131`, `140`. Utilizând acelaşi algoritm, se generează toate numerele naturale din intervalul `[1000,9999]` care au suma cifrelor egală cu `6`. Al treilea număr generat este: **(4p.)**

**a.** `1005` **b.** `1023` **c.** `1031` **d.** `1041`

**Scrieţi pe foaia de examen răspunsul pentru fiecare dintre cerinţele următoare.**

**2.** Subprogramul `F` este definit alăturat. Scrieţi ce se afişează în urma apelului de mai jos.

`F('d');` **(6p.)**

```
void F(char c)
{ if(c>='a')
   { cout<<c; | printf("%c",c);
     F(c-1);
   }
}
```

**3.** Șirul lui Fibonacci (`1`, `1`, `2`, `3`, `5`, `8`, `13`, `21`,…) se definește astfel: $f_1=1$, $f_2=1$ și $f_i=f_{i-1}+f_{i-2}$ pentru orice număr natural $i$, $i\geq 3$.
Subprogramul `Fibo` are un singur parametru, `n`, prin care primeşte un număr natural ($n\in[1,30]$). Subprogramul returnează al `n`-lea termen impar al șirului lui Fibonacci.
Scrieţi definiţia completă a subprogramului.
**Exemplu:** dacă `n=6`, subprogramul returnează numărul `21`. **(10p.)**

**4.** Fișierul `bac.txt` conține un șir de cel mult un milion de numere naturale din intervalul $[0,10^2]$, separate prin câte un spațiu.
Se cere să se determine toate perechile distincte formate din termeni ai șirului aflat în fișier, `x` și `y` ($y-x\geq 2$), astfel încât să nu existe niciun termen al șirului care să aparțină intervalului `(x,y)`. Numerele din fiecare pereche sunt afișate pe câte o linie a ecranului, în ordine strict crescătoare, separate printr-un spațiu, iar dacă nu există nicio astfel de pereche, se afișează pe ecran mesajul `nu exista`. Pentru determinarea numerelor cerute utilizați un algoritm eficient din punctul de vedere al timpului de executare.
**Exemplu:** dacă fișierul conține numerele
`5 9 0 8 10 11 12 13 15 14 6 7 40 10 0 0 5 41 95 7`
atunci pe ecran se afișează, nu neapărat în această ordine, perechile
`0 5`
`15 40`
`41 95`
**a)** Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia. **(4p.)**
**b)** Scrieți programul C/C++ corespunzător algoritmului descris. **(6p.)**

## 19. Informatica — Subiectul III, exercițiul 1

**Sursă: 2016 · Model Oficial**

*2016_Informatica_Mate-Info_Model_Oficial_C-Cpp_Subiect.pdf*

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

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

**1.** Subprogramul `F` este incomplet definit alăturat. Indicați expresia cu care pot fi înlocuite punctele de suspensie, astfel încât, în urma apelului de mai jos, să se afișeze `dcba`.
`F('a');` **(4p.)**

```
void F(char c)
{ if(c<'e')
  { F(......);
    cout<<c; | printf("%c",c);
  }
}
```

**a.** `c-4` **b.** `c-1` **c.** `c+1` **d.** `c+4`

**Scrieţi pe foaia de examen răspunsul pentru fiecare dintre cerinţele următoare.**

**2.** Utilizând metoda backtracking, se generează toate parfumurile formate prin amestecarea a câte `3` esențe distincte din mulţimea {**ambră**, **cedru**, **iris**, **mosc**, **santal**}. Primele cinci soluţii obţinute sunt, în această ordine: (**ambră**, **cedru**, **iris**), (**ambră**, **cedru**, **mosc**), (**ambră**, **cedru**, **santal**), (**ambră**, **iris**, **mosc**) şi (**ambră**, **iris**, **santal**). Scrieți a șasea și a șaptea soluție, în ordinea generării acestora. **(6p.)**

**3.** Subprogramul `Cifre` are un singur parametru, `n`, prin care primește un număr natural ($n \in [0, 10^9]$). Subprogramul returnează numărul cifrelor care apar o singură dată în scrierea lui `n`. Scrieţi definiţia completă a subprogramului.
**Exemplu:** dacă `n=9272017`, subprogramul returnează valoarea `3`. **(10p.)**

**4.** Fişierul `date.in` conţine un şir de cel mult un milion de numere naturale din intervalul $[0, 10^9]$, separate prin câte un spațiu. Șirul are cel puțin doi termeni pari și cel puțin doi termeni impari.
Se cere să se afișeze pe ecran mesajul `DA` dacă șirul aflat în fișier are un subșir ordonat strict crescător, format din toți termenii impari ai săi, și un subșir ordonat strict descrescător, format din toți termenii pari ai săi. Dacă nu există două astfel de subșiruri, programul afișează pe ecran mesajul `NU`. Pentru verificarea proprietății cerute utilizați un algoritm eficient din punctul de vedere al timpului de executare și al memoriei necesare.
**Exemplu:** dacă fişierul conţine numerele
`8 1 6 3 5 4 7`
se afişează pe ecran mesajul
`DA`
iar dacă fișierul conține numerele
`2 1 6 3 5 4 7`
se afişează pe ecran mesajul
`NU`
**a)** Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia. **(4p.)**
**b)** Scrieți programul C/C++ corespunzător algoritmului descris. **(6p.)**

## 20. Informatica — Subiectul III, exercițiul 1

**Sursă: 2016 · Simulare Nationala**

*2016_Informatica_Mate-Info_Simulare_Nationala_C-Cpp_Subiect.pdf*

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

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

**1.** Se consideră subprogramul `f`, definit alăturat. Indicați numărul valorilor întregi afișate în urma apelului de mai jos.
`f(2024,2);`
**(4p.)**

```
void f(int x, int nr)
{ if(x*nr>0)
  { if(x%4==0)
      cout<<x<<' '; | printf("%d ",x);
    f(x-4,nr-1);
  }
  cout<<x<<' '; | printf("%d ",x);
}
```

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

**Scrieţi pe foaia de examen răspunsul pentru fiecare dintre cerinţele următoare.**

**2.** Utilizând metoda backtracking se generează, în ordine lexicografică, toate şirurile de `4` litere distincte din mulţimea {`s`, `i`, `m`, `u`, `l`, `a`, `r`, `e`}, în care nu există două vocale sau două consoane alăturate. Primele `5` şiruri generate sunt, în această ordine: `alem`, `aler`, `ales`, `alim`, `alir`. Scrieți, în ordinea obținerii lor, următoarele două șiruri generate imediat după şirul `ruse`. **(6p.)**

**3.** Un număr natural nenul se numeşte **sPar** dacă atât el, cât și suma divizorilor săi proprii (divizori diferiţi de `1` şi de el însuși), sunt numere pare.
Subprogramul `sPar` are un singur parametru, `n`, prin care primeşte un număr natural ($n \in [1,10^9]$). Subprogramul returnează cel mai mic număr sPar, strict mai mare decât `n`.
Scrieţi definiţia completă a subprogramului.
**Exemplu:** dacă `n=95`, atunci subprogramul returnează `98` (atât `98`, cât și `72=2+7+14+49`, sunt numere pare). **(10p.)**

**4.** Un număr `x` se numește **sufix** al unui număr `y` dacă `y` se poate obține din `x` prin alipirea, la stânga sa, a cel puțin unei alte cifre.
Fişierul `bac.in` conţine un şir de cel mult $10^9$ numere naturale din intervalul $[0,10^9]$. Numerele din şir sunt separate prin câte un spaţiu.
Se cere să se afișeze pe ecran, în ordine strict crescătoare, toate numerele din intervalul `[100,999]` care sunt termeni ai șirului aflat în fișier și sunt sufixe pentru cel puțin un alt termen al aceluiași șir. Numerele sunt afișate câte unul pe linie, iar dacă în șir nu există astfel de numere, se afișează pe ecran mesajul `Nu exista`. Pentru determinarea numerelor cerute se utilizează un algoritm eficient din punctul de vedere al timpului de executare.
**Exemplu:** dacă fişierul `bac.in` are conţinutul
`15502 49 54321 6149 76149 123 123 502 4321 321 321`
atunci pe ecran se afişează numerele
`321`
`502`
**a)** Descrieţi în limbaj natural algoritmul utilizat, justificând eficienţa acestuia. **(4p.)**
**b)** Scrieţi programul C/C++ corespunzător algoritmului descris. **(6p.)**

## 21. Informatica — Subiectul III, exercițiul 1

**Sursă: 2016 · Speciala C · V04**

*2016_Informatica_Mate-Info_Sesiune_Speciala_C-Cpp_Subiect_V04.pdf*

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

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

**1.** Subprogramul `f` este definit alăturat. Indicați cea mai mare valoare din intervalul `[20,25]` pe care o poate avea variabila întreagă `x`, astfel încât, în urma apelului de mai jos, să se afișeze numerele `2 4 16`.

`f(x,1);` **(4p.)**

```
void f(int n, int p)
{ if(n>0)
  { if(n%2==1)
      cout<<p<<" "; | printf(" ",p);
    f(n/2, p*2);
  }
}
```

**a.** `21` **b.** `22` **c.** `23` **d.** `24`

**Scrieţi pe foaia de examen răspunsul pentru fiecare dintre cerinţele următoare.**

**2.** Utilizând metoda backtracking se generează, în ordine lexicografică, toate şirurile de câte `6` cifre din mulţimea {`0`,`1`} cu proprietatea că au cel mult două cifre cu valori egale pe poziţii consecutive. Primele `5` soluții generate sunt, în această ordine: `001001`, `001010`, `001011`, `001100`, `001101`. Scrieți a 7-a și a 8-a soluție, în ordinea generării acestora. **(6p.)**

**3.** Subprogramul `minDivPrim` are un singur parametru, `n`, prin care primeşte un număr natural (`n`$\in$`[2,10^9]`). Subprogramul returnează cel mai mic număr natural care are aceiași divizori primi ca `n`.
Scrieţi definiţia completă a subprogramului.
**Exemplu:** dacă `n=75`, subprogramul returnează numărul `15`, iar dacă `n=7`, subprogramul returnează numărul `7`. **(10p.)**

**4.** Se consideră șirul `1`, `-1`, `2 ...` definit astfel: $f_1=1$, $f_2=-1$, iar $f_n=1-2\cdot f_{n-1}-f_{n-2}$, dacă $n\geq 3$ (unde `n` este un număr natural).
Se citește de la tastatură un număr natural, `n` (`n`$\in$`[1,10^9]`), și se cere să se scrie în fișierul text `bac.out`, separați prin câte un spațiu, primii `n` termeni ai șirului, în ordine inversă apariției lor în acesta.
Pentru determinarea și afișarea numerelor cerute se utilizează un algoritm eficient din punctul de vedere al spațiului de memorie și al timpului de executare.
**Exemplu:** dacă `n=3`, fişierul conţine valorile
`2 -1 1`
**a)** Descrieţi în limbaj natural algoritmul utilizat, justificând eficienţa acestuia. **(4p.)**
**b)** Scrieţi programul C/C++ corespunzător algoritmului descris. **(6p.)**

## 22. Informatica — Subiectul III, exercițiul 1

**Sursă: 2016 · Toamna August · V09**

*2016_Informatica_Mate-Info_Sesiune_Toamna_August-Septembrie_C-Cpp_Subiect_V09.pdf*

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

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

**1.** Subprogramele **f1** şi **f2** sunt definite mai jos.

```
int f1(int n)
{ if(n==0) return 1;
  else return n*f1(n-1);
}
```

```
int f2(int n)
{ if(n!=1) return n*f2(n-1);
  else return 0;
}
```

La apel, pentru parametrul **n=5**, returnează valoarea **120**: **(4p.)**

**a.** atât **f1**, cât şi **f2** **b.** numai **f1** **c.** numai **f2** **d.** nici **f1**, nici **f2**

**Scrieţi pe foaia de examen răspunsul pentru fiecare dintre cerinţele următoare.**

**2.** Utilizând metoda bactracking se generează în ordine lexicografică toate şirurile de **5** litere distincte din mulţimea {**C**, **A**, **R**, **T**, **E**}, astfel încât în fiecare şir litera **T** precede litera **A**. Primele trei soluţii generate sunt, în această ordine: **CERTA**, **CETAR**, **CETRA**. Scrieţi cea de a patra şi cea de a cincea soluţie, în ordinea generării acestora. **(6p.)**

**3.** Subprogramul **duplicare** are doi parametri:

- **n**, prin care primeşte un număr natural ($n \in [1, 10^4]$);
- **d**, prin care furnizează numărul obţinut prin duplicarea fiecărei cifre pare a lui **n** sau **-1** dacă acesta nu are nicio cifră pară.

Scrieţi definiţia completă a subprogramului.
**Exemplu:** dacă **n=2380**, după apel **d=2238800**. **(10p.)**

**4.** Fişierul **bac.txt** conţine un şir de cel mult $10^6$ numere naturale distincte din intervalul $[0, 10^9]$. Numerele din şir sunt separate prin câte un spaţiu.
Se cere să se determine cei mai mari doi termeni pari din şir care sunt precedaţi de doar trei termeni impari. Termenii determinați se afișează pe ecran, în ordine strict crescătoare, separați printr-un spațiu, iar dacă în şir nu există doi astfel de termeni, se afişează pe ecran mesajul **Nu exista**.
Pentru determinarea termenilor ceruţi se utilizează un algoritm eficient din punctul de vedere al spațiului de memorie şi al timpului de executare.
**Exemplu:** dacă fişierul conţine numerele
**4 3 5 312 27 30 14 212 11 15 17 400**
se afişează pe ecran numerele
**30 212**
(**4** este precedat de **0** numere impare, **312** este precedat de **2** numere impare, **30**, **14** și **212** sunt precedate de **3** numere impare, iar **400** este precedat de **6** numere impare; dintre numerele **30**, **14** și **212** cele mai mari sunt **30** și **212**).
**a)** Descrieţi în limbaj natural algoritmul utilizat, justificând eficienţa acestuia. **(4p.)**
**b)** Scrieţi programul C/C++ corespunzător algoritmului descris. **(6p.)**

## 23. Informatica — Subiectul III, exercițiul 1

**Sursă: 2016 · Vara Iunie · V10**

*2016_Informatica_Mate-Info_Sesiune_Vara_Iunie-Iulie_C-Cpp_Subiect_V10.pdf*

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

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

**1.** Subprogramul `f` este definit alăturat. Indicaţi ce se afişează în urma apelului de mai jos.
`f(2016);` **(4p.)**

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

**a.** `6102` **b.** `2202012016` **c.** `2016201202` **d.** `20162012020`

**Scrieţi pe foaia de examen răspunsul pentru fiecare dintre cerinţele următoare.**

**2.** Având la dispoziţie cinci tipuri de prăjituri, cu **`pere`**, cu **`mure`**, cu **`afine`**, cu **`fragi`**, cu **`zmeură`**, se utilizează metoda backtracking pentru a obţine toate posibilităţile de a forma platouri cu câte trei tipuri de prăjituri diferite, ştiind că în cadrul unui platou nu contează ordinea de aşezare a prăjiturilor şi că prăjiturile cu **`mure`** nu vor fi plasate pe acelaşi platou cu prăjiturile cu **`fragi`**. Primele patru soluţii obţinute sunt, în această ordine: (**`pere`**, **`mure`**, **`afine`**), (**`pere`**, **`mure`**, **`zmeură`**), (**`pere`**, **`afine`**, **`fragi`**), (**`pere`**, **`afine`**, **`zmeură`**). Scrieţi cea de a cincea şi cea de a şasea soluţie, în ordinea obţinerii lor. **(6p.)**

**3.** Subprogramul **`cifreImpare`** are un singur parametru, **`n`**, prin care primeşte un număr natural cu toate cifrele nenule (**`n`**$\in$ **`[1,10`**$^{\mathbf{9}}$**`]`**). Subprogramul returnează numărul obţinut prin eliminarea tuturor cifrelor impare din **`n`**, respectiv **`-1`** dacă nu există astfel de cifre sau dacă toate cifrele lui **`n`** sunt impare.
Scrieţi definiţia completă a subprogramului.
**Exemplu:** dacă **`n=23541`** subprogramul returnează **`24`**, iar dacă **`n=28`** subprogramul returnează **`-1`**. **(10p.)**

**4.** Fişierul **`numere.in`** conţine pe prima linie un număr natural **`n`** (**`n`**$\in$ **`[2,10`**$^{\mathbf{9}}$**`]`**), iar pe a doua linie un şir de cel mult **`10`**$^{\mathbf{9}}$ numere naturale din intervalul **`[1,n]`**. Numerele din şir sunt ordonate crescător şi sunt separate prin câte un spaţiu.
Se cere să se determine valorile naturale distincte din intervalul **`[1,n]`** care **NU** se găsesc în şirul menţionat mai sus. Valorile determinate se afişează pe ecran în ordine strict crescătoare, separate prin câte un spaţiu. Dacă nu există astfel de valori, se afişează pe ecran mesajul **`Nu exista`**.
Pentru determinarea valorilor cerute se utilizează un algoritm eficient din punctul de vedere al memoriei şi al timpului de executare.
**Exemplu:** dacă fişierul conţine numerele
`10`
`3 4 4 8`
se afişează pe ecran valorile
`1 2 5 6 7 9 10`
**a)** Descrieţi în limbaj natural algoritmul utilizat, justificând eficienţa acestuia. **(4p.)**
**b)** Scrieţi programul C/C++ corespunzător algoritmului descris. **(6p.)**

## 24. Informatica — Subiectul III, exercițiul 1

**Sursă: 2017 · Model Oficial**

*2017_Informatica_Mate-Info_Model_Oficial_C-Cpp_Subiect.pdf*

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

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

**1.** Utilizând metoda bactracking se generează toate submulţimile cu cel mult patru instrumente muzicale din mulţimea {**clarinet**, **corn**, **flaut**, **oboi**, **saxofon**}. Primele șase soluţii generate sunt, în această ordine: {**clarinet**}, {**clarinet**, **corn**}, {**clarinet**, **corn**, **flaut**}, {**clarinet**, **corn**, **flaut**, **oboi**}, {**clarinet**, **corn**, **flaut**, **saxofon**}, {**clarinet**, **corn**, **oboi**}. Cea de a opta soluţie este:

**a.** {**corn**} **b.** {**clarinet**, **flaut**}

**c.** {**clarinet**, **corn**, **saxofon**} **d.** {**clarinet**, **corn**, **oboi**, **saxofon**}

**Scrieţi pe foaia de examen răspunsul pentru fiecare dintre cerinţele următoare.**

**2.** Subprogramul **f** este definit alăturat. Scrieţi ce se afişează în urma apelului de mai jos.

```
f(12);
```

**(6p.)**

```
void f (int n)
{ int i;
  for(i=2;i<=n/2;i++)
    if(n%i==0)
    {
      cout<<i<<' '; | printf("%d ",i);
      f (n/i);
    }
}
```

**3.** Subprogramul **nrDiv** are doi parametri, **a** și **b** (**a≤b**), prin care primeşte câte un număr natural din intervalul $[1,10^9]$. Subprogramul returnează numărul valorilor din intervalul **[a,b]** care pot fi scrise ca produs de două numere naturale consecutive.
Scrieţi definiţia completă a subprogramului.
**Exemplu:** dacă **a=10** și **b=40**, subprogramul returnează **3** (valorile cu proprietatea cerută sunt **12**, **20** și **30**). **(10p.)**

**4.** Se consideră şirul definit alăturat (unde **n** și **x** sunt numere naturale nenule, iar **x** este impar). De exemplu, pentru **x=21** șirul este:
**21**, **22**, **43**, **44**, **87**, **88**, **175**, **176** ....

$$f_n = \begin{cases} x, & \text{dacă } n = 1 \\ 1 + f_{n-1}, & \text{dacă } n \text{ par} \\ 1 + 2 \cdot f_{n-2}, & \text{altfel} \end{cases}$$

Se citesc de la tastatură două numere naturale din intervalul $[1,10^9]$, **x** și **y**, cu cel mult nouă cifre, unde **x** are semnificația precizată mai sus, iar **y** este un termen al şirului dat, şi se cere să se scrie în fişierul text **bac.txt**, în ordine strict descrescătoare, separați prin câte un spațiu, toţi termenii şirului care sunt mai mici sau egali cu **y**.
Pentru determinarea termenilor ceruţi se utilizează un algoritm eficient din punctul de vedere al memoriei şi al timpului de executare.
**Exemplu:** dacă **x=21**, iar **y=175**, fişierul **bac.txt** conţine numerele
**175 88 87 44 43 22 21**
**a)** Descrieți în limbaj natural algoritmul utilizat, justificând eficienţa acestuia. **(2p.)**
**b)** Scrieţi programul C/C++ corespunzător algoritmului descris. **(8p.)**

## 25. Informatica — Subiectul III, exercițiul 1

**Sursă: 2017 · Simulare Nationala**

*2017_Informatica_Mate-Info_Simulare_Nationala_C-Cpp_Subiect.pdf*

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

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

**1.** Subprogramul **f** este definit alăturat. Indicați ce se afișează în urma apelului de mai jos.
**f(6);**
**(4p.)**

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

**a.** 642 **b.** 6446 **c.** 64246 **d.** 642246

**Scrieţi pe foaia de examen răspunsul pentru fiecare dintre cerinţele următoare.**

**2.** Se utilizează metoda backtracking pentru a obține toate buchetele formate din câte trei tipuri de flori de primăvară din mulțimea {**brânduș**ă, **iasomie**, **lalea**, **liliac**, **margaretă**}, astfel încât iasomia și liliacul nu vor fi plasate în același buchet. Știind că în cadrul unui buchet nu contează ordinea de aşezare a florilor, primele patru soluţii obţinute sunt, în această ordine: (**brânduș**ă, **iasomie**, **lalea**), (**brânduș**ă, **iasomie**, **margaretă**), (**brânduș**ă, **lalea**, **liliac**), (**brânduș**ă, **lalea**, **margaretă**). Scrieţi cea de a cincea şi cea de a şasea soluţie, în ordinea obţinerii lor. **(6p.)**

**3.** Subprogramul **ranguri** are un singur parametru, **n**, prin care primeşte un număr natural ($n \in [0, 10^9]$). Subprogramul returnează numărul de cifre ale lui **n** care sunt egale cu pozițiile pe care le ocupă în scrierea acestuia. Pozițiile sunt numerotate de la dreapta la stânga, iar cifra unităților ocupă poziția **0**.
Scrieţi definiţia completă a subprogramului.
**Exemplu:** dacă **n=659427**0, subprogramul returnează numărul **4**. **(10p.)**

**4.** Fișierul **bac.in** conține pe prima linie două numere naturale din intervalul $[2, 10^4]$, **m** și **n**, iar pe fiecare dintre următoarele două linii câte un șir de **m**, respectiv **n** numere naturale din intervalul $[0, 10^9]$, ordonate **strict crescător**. Numerele aflate pe aceeași linie a fișierului sunt separate prin câte un spațiu.
Se cere să se afișeze pe ecran, în ordine **strict descrescătoare**, numerele pare care apar în cel puțin unul dintre cele două șiruri. Numerele afișate sunt separate prin câte un spațiu, iar dacă nu există niciun astfel de număr, se afișează pe ecran mesajul **nu exista**.
Pentru determinarea numerelor cerute se va utiliza un algoritm eficient din punctul de vedere al timpului de executare.
**Exemplu:** dacă fișierul conține numerele
**5 6**
**1 4 8 9 10**
**2 4 10 12 15 18**
se afișează pe ecran
**18 12 10 8 4 2**
**a)** Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia. **(2p.)**
**b)** Scrieţi programul C/C++ corespunzător algoritmului descris. **(8p.)**

## 26. Informatica — Subiectul III, exercițiul 1

**Sursă: 2017 · Speciala C · V03**

*2017_Informatica_Mate-Info_Sesiune_Speciala_C-Cpp_Subiect_V03.pdf*

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

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

**1.** Utilizând metoda backtracking se generează, în ordine crescătoare, toate numerele naturale pare cu trei cifre, cu proprietatea că nu există două cifre egale alăturate și suma cifrelor este `10`. Primele cinci numere generate sunt, în această ordine: `136`, `154`, `172`, `190`, `208`. Al şaselea număr generat este: **(4p.)**

**a.** `217` **b.** `226` **c.** `262` **d.** `280`

**Scrieţi pe foaia de examen răspunsul pentru fiecare dintre cerinţele următoare.**

**2.** Subprogramul `f` este definit alăturat. Scrieţi ce se afişează în urma apelului de mai jos. **(6p.)**

```
f(4);
```

```
void f(int n)
{ int i;
  if(n>0)
  { for(i=1;i<=n;i++)
      cout<<i;   |  printf("%d",i);
    f(n-1);
  }
}
```

**3.** Subprogramul `identice` are doi parametri, `a` și `b`, prin care primește câte un număr natural ($10 \le a \le b \le 10^6$). Subprogramul afișează pe ecran toate numerele naturale din intervalul `[a,b]` care au toate cifrele identice. Numerele afișate sunt separate prin câte un spațiu, iar dacă nu există astfel de numere, se afişează pe ecran mesajul `nu exista`.
Scrieţi definiţia completă a subprogramului.
**Exemplu**: pentru `a=700` și `b=1500` se afișează pe ecran
`777 888 999 1111`. **(10p.)**

**4.** Numim **inserare** a unui șir `A` într-un șir `B` introducerea, între două elemente ale șirului `B`, a tuturor elementelor lui `A`, pe poziții consecutive, în ordinea în care apar în `A`.
Fișierul `bac.in` conține numere naturale din intervalul $[1,10^6]$: pe prima linie numerele `m` și `n`, iar pe fiecare dintre următoarele două linii câte un șir de `m`, respectiv de `n` numere întregi ordonate **strict crescător**. Numerele aflate pe aceeași linie a fișierului sunt separate prin câte un spațiu, iar numerotarea elementelor în șiruri începe de la `1`.
Se cere să se afișeze pe ecran poziția din al doilea șir începând de la care poate fi inserat primul șir, astfel încât șirul obținut să fie strict crescător. Dacă nu există o astfel de poziție, se afișează pe ecran mesajul `imposibil`.
Proiectați un algoritm eficient din punctul de vedere al spațiului de memorie utilizat şi al timpului de executare.
**Exemplu:** dacă fișierul conține numerele

```
4 6
15 16 17 19
7 10 12 20 30 40
```

se poate obține șirul `7`, `10`, `12`, `15`, `16`, `17`, `19`, `20`, `30`, `40` și se afișează pe ecran `4`
iar dacă fișierul conține numerele

```
4 6
15 16 17 19
7 14 18 20 30 40
```

sau numerele

```
4 6
1 2 3 4
7 15 18 20 30 40
```

se afișează pe ecran mesajul `imposibil`
**a)** Descrieţi în limbaj natural algoritmul proiectat, justificând eficienţa acestuia. **(2p.)**
**b)** Scrieţi programul C/C++ corespunzător algoritmului descris. **(8p.)**

## 27. Informatica — Subiectul III, exercițiul 1

**Sursă: 2017 · Toamna August · V07**

*2017_Informatica_Mate-Info_Sesiune_Toamna_August-Septembrie_C-Cpp_Subiect_V07.pdf*

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

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

**1.** Subprogramul `f` este definit alăturat. Valoarea lui `f(1234)` este: **(4p.)**

```
int f(int n)
{ if (n!=0)
    return (n%2)*(n%10)+f(n/10);
  return 0;
}
```

**a.** `0` **b.** `4` **c.** `6` **d.** `10`

**Scrieţi pe foaia de examen răspunsul pentru fiecare dintre cerinţele următoare.**

**2.** Fiind date o mulțime finită și un set de perechi de forma `(i,j)`, unde `i` și `j` sunt elemente ale mulțimii, se numește **șir sortat topologic asociat mulțimii** un șir format din toate elementele mulțimii, astfel încât pentru orice pereche `(i,j)` dintre cele date, elementul `i` să se găsească înaintea lui `j` în şir.
Utilizând metoda backtracking, se generează toate șirurile sortate topologic ale mulţimii {`1`, `2`, `3`, `4`, `5`, `6`}, având în vedere perechile `(4,6)`, `(3,4)`, `(3,1)`, `(2,5)`. Primele trei soluţii generate sunt, în această ordine: `(2,3,1,4,5,6)`, `(2,3,1,4,6,5)`, `(2,3,1,5,4,6)`. Scrieţi cea de a patra şi cea de a cincea soluţie, în ordinea generării acestora. **(6p.)**

**3.** Subprogramul `triplete` are trei parametri, `a`, `b` și `c`, prin care primește câte un număr natural din intervalul $[0,10^4]$ (`a<b`). Subprogramul afișează pe ecran toate tripletele de numere naturale, `x`, `y` şi `z`, soluţii ale ecuaţiei `x+y+z=c`, cu proprietatea că $a \le x \le y \le z \le b$. Fiecare triplet afișat este încadrat între acolade, iar numerele `x`, `y` și `z` sunt afișate în această ordine, separate prin câte o virgulă. Dacă nu există niciun triplet cu proprietatea cerută, se afişează mesajul `nu exista`.
Scrieți definiția completă a subprogramului.
**Exemplu**: dacă `a=2`, `b=4` şi `c=8`, se afișează pe ecran tripletele
`{2,2,4}{2,3,3}`
iar dacă `a=5`, `b=8`, `c=8` se afişează pe ecran mesajul
`nu exista`. **(10p.)**

**4.** Un șir de numere este o **progresie aritmetică de rație** `r` dacă oricare termen al său, cu excepția primului, se obține din cel care îl precede, prin adunarea la acesta a numărului `r`.
**Exemplu:** șirul `12`, `14`, `16`, `18`, `20` este o progresie de rație `2`.
Fișierul `bac.in` conține un șir de cel mult $10^6$ numere naturale din intervalul $[0,10^3]$, separate prin câte un spațiu.
Se cere să se verifice dacă există un număr natural `r`, astfel încât toate numerele **distincte** din șir să poată fi rearanjate, pentru a forma o progresie aritmetică de rație `r`. Se afișează pe ecran numărul `r`, sau mesajul `NU`, dacă nu există un astfel de număr. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.
**Exemplu:** dacă fișierul conține numerele
`180 30 80 280 130 330 230 30 30 330 80`
se afișează pe ecran
`50`
**a)** Descrieţi în limbaj natural algoritmul proiectat, justificând eficienţa acestuia. **(2p.)**
**b)** Scrieţi programul C/C++ corespunzător algoritmului descris. **(8p.)**

## 28. Informatica — Subiectul III, exercițiul 1

**Sursă: 2017 · Vara Iunie · V04**

*2017_Informatica_Mate-Info_Sesiune_Vara_Iunie-Iulie_C-Cpp_Subiect_V04.pdf*

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

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

**1.** Utilizând metoda backtracking se generează, în ordine strict crescătoare, toate numerele de trei cifre din mulţimea {**1**, **2**, **5**, **7**, **8**}, numere cu proprietatea că au cel mult două cifre impare. Primele şapte numere generate sunt, în această ordine: **112**, **118**, **121**, **122**, **125**, **127**, **128**. Al optulea număr generat este: **(4p.)**

**a.** 151 **b.** 152 **c.** 157 **d.** 158

**Scrieţi pe foaia de examen răspunsul pentru fiecare dintre cerinţele următoare.**

**2.** Subprogramul **f** este definit alăturat. Scrieţi valorile **f(3,9)** şi **f(1,1000)**. **(6p.)**

```
int f(int x,int y)
{ if(x*5>y/5)
    return x;
  return f(x*5,y/5);
}
```

**3.** Subprogramul **duplicare** are un singur parametru, **n**, prin care primește un număr natural ($n \in [1, 10^4)$). Subprogramul furnizează, prin același parametru, numărul obţinut din **n** prin inserarea, după fiecare cifră pară din scrierea lui, a unei cifre egale cu jumătate din aceasta.
Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **n=2380** după apel, **n=2138400**, iar dacă **n=35** după apel, **n=35**. **(10p.)**

**4.** Numim **secvență pară** într-un șir o succesiune de termeni ai șirului cu proprietatea că sunt numere pare și că se află pe poziții consecutive în șir; orice secvență are cel puțin doi termeni și este maximală în raport cu proprietatea precizată (dacă i se adaugă un alt termen, secvența își pierde această proprietate). Lungimea secvenței este egală cu numărul termenilor săi.
Fişierul **bac.txt** conţine un şir de cel mult $10^6$ numere naturale din intervalul $[0, 10^9]$. Numerele din şir sunt separate prin câte un spaţiu.
Se cere să se afişeze pe ecran numărul de secvențe pare de lungime maximă din șir.
Proiectați un algoritm eficient din punctul de vedere al spațiului de memorie utilizat şi al timpului de executare.
**Exemplu:** dacă fişierul **bac.txt** conţine valorile
**1 2 3 4 6 10 2 8 5 7 9 4 6 10 121 20 4 11 10 2 5 2 6 8 10 16**
se afişează pe ecran numărul **2**.
**a)** Descrieţi în limbaj natural algoritmul proiectat, justificând eficienţa acestuia. **(2p.)**
**b)** Scrieţi programul C/C++ corespunzător algoritmului descris. **(8p.)**

## 29. Informatica — Subiectul III, exercițiul 1

**Sursă: 2018 · Model Oficial**

*2018_Informatica_Mate-Info_Model_Oficial_C-Cpp_Subiect.pdf*

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

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

**1.** Se consideră subprogramul `f`, definit alăturat. Valoarea `f(20,18)` este: **(4p.)**

```
int f(int x, int y)
{ if (x==y) return 0;
  if(x>y) return 1+f(x/2,y);
  return 1+f(x,y/2);
}
```

**a.** 2 **b.** 4 **c.** 6 **d.** 8

**Scrieţi pe foaia de examen răspunsul pentru fiecare dintre cerinţele următoare.**

**2.** Utilizând metoda backtracking, se generează toate posibilitățile de a forma succesiuni de câte `5` genuri muzicale distincte din mulțimea {**baladă**, **doină**, **fado**, **romanță**, **trova**}, astfel încât în fiecare succesiune genul fado precede genul romanță. Două succesiuni sunt distincte dacă genurile muzicale sunt în altă ordine.
Primele cinci soluţii generate sunt, în această ordine, (**baladă**, **doină**, **fado**, **romanță**, **trova**), (**baladă**, **doină**, **fado**, **trova**, **romanță**), (**baladă**, **doină**, **trova**, **fado**, **romanță**), (**baladă**, **fado**, **doină**, **romanță**, **trova**), (**baladă**, **fado**, **doină**, **trova**, **romanță**). Scrieți, în ordinea obținerii, ultimele două soluții generate imediat înainte de (**trova**, **fado**, **romanță**, **baladă**, **doină**). **(6p.)**

**3.** Se consideră subprogramul **`divizori`**, cu patru parametri:
- `n`, prin care primeşte un număr natural ($n \in [2, 10^6]$);
- `x`, `y` și `z`, prin care furnizează câte un divizor al lui `n` (`0<x<y<z`) cu proprietatea că `x+y+z=n`, sau valoarea `0`, prin fiecare dintre aceștia, dacă `n` nu are astfel de divizori.

Scrieţi definiţia completă a subprogramului.
**Exemplu**: pentru numărul `n=24`, în urma apelului, `x=4`, `y=8` şi `z=12`, iar pentru numărul `n=9`, în urma apelului, `x=0`, `y=0` şi `z=0`. **(10p.)**

**4.** Un număr natural `x`, format din exact două cifre, este numit **sub-număr** al unui număr natural `y` dacă cifrele lui `x` apar, în aceeași ordine, pe ranguri consecutive, în numărul `y`.
**Exemplu:** `21` este sub-număr al lui `12145`, al lui `213`, al lui `21`, dar nu și al lui `123` sau al lui `231`.
Fișierul **`bac.txt`** conține cel mult $10^6$ numere naturale din intervalul [`10`, $10^9$], separate prin câte un spațiu.
Se cere să se afișeze pe ecran, separate prin câte un spațiu, sub-numerele care apar o singură dată în scrierea numerelor din fișier, sau mesajul **`nu exista`** dacă nu există niciun astfel de sub-număr. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.
**Exemplu:** dacă fişierul **`bac.txt`** conţine numerele
`391 7772 4358 23972`
atunci pe ecran se afișează valorile de mai jos, nu neapărat în această ordine:
`91 43 35 58 23 97`
**a)** Descrieţi în limbaj natural algoritmul proiectat, justificând eficienţa acestuia. **(2p.)**
**b)** Scrieţi programul C/C++ corespunzător algoritmului descris. **(8p.)**

## 30. Informatica — Subiectul III, exercițiul 1

**Sursă: 2018 · Simulare Nationala**

*2018_Informatica_Mate-Info_Simulare_Nationala_C-Cpp_Subiect.pdf*

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

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

**1.** Subprogramul `f` este definit alăturat. Valoarea `f(2203,2018)` este: **(4p.)**

```
int f(int x, int y)
{ if(x*y==0) return 0;
  if(x%2==y%2) return 1+10*f(x/10,y/10);
  return 10*f(x/10,y/10);
}
```

**a.** `1100` **b.** `11` **c.** `2` **d.** `0`

**Scrieţi pe foaia de examen răspunsul pentru fiecare dintre cerinţele următoare.**

**2.** Utilizând metoda backtracking, se generează toate posibilitățile de a forma șiraguri de câte `4` mărgele de culori distincte din mulţimea {**roșu**, **albastru**, **roz**, **portocaliu**, **verde**}, astfel încât în fiecare șirag nu pot fi pe poziții alăturate mărgele roșii și albastre. Două șiraguri sunt distincte dacă au cel puțin o mărgea de culoare diferită sau dacă ordinea culorilor mărgelelor este diferită.
Primele cinci soluţii generate sunt, în această ordine, (**roșu**, **roz**, **albastru**, **portocaliu**), (**roșu**, **roz**, **albastru**, **verde**), (**roșu**, **roz**, **portocaliu**, **albastru**), (**roșu**, **roz**, **portocaliu**, **verde**), (**roșu**, **roz**, **verde**, **albastru**). Scrieţi cea de a şasea şi cea de a şaptea soluţie, în ordinea generării acestora. **(6p.)**

**3.** Se consideră subprogramul `radical`, cu trei parametri:
- `n`, prin care primeşte un număr natural (`n`$\in$ `[1,10^9]`);
- `x` și `y`, prin care furnizează două numere naturale cu proprietatea că $\sqrt{n}$ poate fi scris sub forma $x \cdot \sqrt{y}$, unde `x` are o valoare maximă.

Scrieţi definiţia completă a subprogramului.
**Exemplu**: pentru numărul `n=15000`, în urma apelului, `x=50` şi `y=6`, iar pentru numărul `n=9`, în urma apelului, `x=3` şi `y=1`. **(10p.)**

**4.** Se consideră un șir ai cărui termeni sunt numere naturale nenule, de o singură cifră. Numim **număr asociat** al acestui șir un număr natural format cu termenii șirului, în ordinea în care aceștia apar în șir.
**Exemplu:** numărul asociat șirului `1`, `2`, `5`, `3`, `2` este `12532`.
Fişierul text `bac.txt` conţine numere naturale din intervalul `[1,9]`: pe prima linie două numere, `x` și `y`, iar pe a doua linie un șir de cel puţin trei şi cel mult $10^5$ termeni. Numerele aflate pe aceeași linie a fișierului sunt separate prin câte un spațiu.
Se cere inserarea valorilor `x` și `y` în șirul aflat pe a doua linie fișierului, astfel încât numărul asociat șirului obținut să fie minim. Termenii șirului obținut se afișează pe ecran, separați prin câte un spațiu.
Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.
**Exemplu:** dacă fişierul `bac.txt` conţine numerele
`9 6`
`1 7 5`
atunci, pentru că numerele asociate șirurilor care se pot obține sunt `96175`, `69175`, `61975`, `61795`, `61759`, `91675`, `19675`, `16975`, `16795`, `16759`, `91765`, `19765`, `17965`, `17695`, `17659`, `91756`, `19756`, `17956`, `17596`, `17569`, pe ecran se afişează șirul:
`1 6 7 5 9`
**a)** Descrieţi în limbaj natural algoritmul proiectat, justificând eficienţa acestuia. **(2p.)**
**b)** Scrieţi programul C/C++ corespunzător algoritmului descris. **(8p.)**

## 31. Informatica — Subiectul III, exercițiul 1

**Sursă: 2018 · Speciala C · V09**

*2018_Informatica_Mate-Info_Sesiune_Speciala_C-Cpp_Subiect_V09.pdf*

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

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

**1.** Subprogramele **f1** şi **f2** sunt definite mai jos.

```
int f1 (int x, int y)
{ if(x%2!=0 || y%2!=0)return 1;
  else return 2*f1(x/2,y/2);
}
```

```
int f2 (int x, int y)
{ if (x==y) return x;
  else
    if(x>y) return f2(x-y,y);
    else return f2(x, y-x);
}
```

Cel mai mare divizor comun al lui **30** și **50** se obține în urma apelului: **(4p.)**

**a.** `f1(30,50)` **b.** `f2(30,50)` **c.** `f1(30/2,50)` **d.** `f2(30/2,50)`

**Scrieţi pe foaia de examen răspunsul pentru fiecare dintre cerinţele următoare.**

**2.** Utilizând metoda backtracking, se generează toate posibilitățile de a forma cutii cu bomboane de tipuri distincte din mulțimea {**fondante**, **caramele**, **dropsuri**, **acadele**}. Într-o cutie sunt cel puțin două tipuri de bomboane, dar nu pot fi și dropsuri și acadele simultan. Două cutii sunt distincte dacă ele conțin cel puțin un tip diferit de bomboane.
Primele patru soluţii generate sunt, în această ordine, (**fondante**, **caramele**), (**fondante**, **caramele**, **dropsuri**), (**fondante**, **caramele**, **acadele**), (**fondante**, **dropsuri**). Scrieţi a cincea și a șasea soluție, în ordinea generării acestora. **(6p.)**

**3.** Un număr natural este numit **echilibrat** dacă suma cifrelor sale de pe poziții pare este un număr par, iar suma cifrelor sale de pe poziţii impare este un număr impar. Pozițiile cifrelor sunt numerotate de la dreapta la stânga, astfel: cifra unităților este pe poziția **0**, cifra zecilor este pe poziția **1** ș.a.m.d.
Subprogramul **echilibrat** are un singur parametru, **n**, prin care primeşte un număr natural (**n**$\in$**[10,10$^9$]**). Subprogramul returnează valoarea **1** dacă **n** este echilibrat sau valoarea **0** în caz contrar.
Scrieţi definiţia completă a subprogramului.
**Exemplu:** dacă **n=25163912**, subprogramul returnează valoarea **1**, iar dacă **n=11211**, subprogramul returnează valoarea **0**. **(10p.)**

**4.** Numim **secvență încadrată** a unui șir de numere naturale un subșir al acestuia, format din termeni aflați pe poziții consecutive în șirul dat, subșir care începe și se termină cu aceeași valoare. Lungimea secvenței este egală cu numărul de termeni ai acesteia.
Fișierul **bac.txt** conține un șir de cel puțin două și cel mult **10$^6$** numere naturale din intervalul [**0**, **9**]. Numerele sunt separate prin câte un spațiu. În șir există cel puțin doi termeni egali.
Se cere să se determine secvențele încadrate din acest șir care au lungime maximă și să se afișeze pe prima linie a ecranului lungimea maximă determinată, iar pe următoarea linie, pentru fiecare astfel de secvenţă, valoarea primului său termen. Numerele de pe a doua linie sunt afișate în ordine strict crescătoare, separate prin câte un spaţiu.
Proiectați un algoritm eficient din punctul de vedere al timpului de executare.
**Exemplu:** dacă fişierul **bac.txt** conţine numerele
**3 1 5 2 4 5 5 2 5 9 5 7 4 6 8 0 8**
atunci pe ecran se afișează valorile:
**9**
**4 5**
**a)** Descrieţi în limbaj natural algoritmul proiectat, justificând eficienţa acestuia. **(2p.)**
**b)** Scrieţi programul C/C++ corespunzător algoritmului descris. **(8p.)**

## 32. Informatica — Subiectul III, exercițiul 1

**Sursă: 2018 · Toamna August · V01**

*2018_Informatica_Mate-Info_Sesiune_Toamna_August-Septembrie_C-Cpp_Subiect_V01.pdf*

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

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

**1.** Subprogramul `f` este definit alăturat. Indicați ce se afișează în urma apelului de mai jos.
`f(7);`
**(4p.)**

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

**a.** `10` **b.** `010` **c.** `101` **d.** `1010`

**Scrieţi pe foaia de examen răspunsul pentru fiecare dintre cerinţele următoare.**

**2.** Utilizând metoda backtracking, se generează toate posibilitățile de a forma seturi de câte `5` instrumente de scris distincte din mulţimea {**`stilou`**, **`pană`**, **`toc`**, **`creion`**, **`pensulă`**}, astfel încât în fiecare set creionul precede stiloul și pana. Două seturi sunt distincte dacă instrumentele sunt dispuse în altă ordine.
Primele cinci soluţii generate sunt, în această ordine, (**`toc`**, **`creion`**, **`stilou`**, **`pană`**, **`pensulă`**), (**`toc`**, **`creion`**, **`stilou`**, **`pensulă`**, **`pană`**), (**`toc`**, **`creion`**, **`pană`**, **`stilou`**, **`pensulă`**), (**`toc`**, **`creion`**, **`pană`**, **`pensulă`**, **`stilou`**), (**`toc`**, **`creion`**, **`pensulă`**, **`stilou`**, **`pană`**). Scrieţi cea de a şasea şi cea de a şaptea soluţie, în ordinea generării acestora.
**(6p.)**

**3.** Subprogramul **`interval`** are un singur parametru, **`n`**, prin care primeşte un număr natural (**`n`**$\in$ **`[3,10^6]`**). Subprogramul returnează cel mai mic număr natural **`x`** (**`n<x`**) care **NU** este prim, cu proprietatea că în intervalul **`[n,x]`** există un singur număr prim.
Scrieţi definiţia completă a subprogramului.
**Exemplu:** dacă **`n=8`**, subprogramul returnează numărul **`12`**. **(10p.)**

**4.** Primii termeni ai șirului definit alăturat (unde **`n`** este un număr natural nenul) sunt:
**`0, 3, 8, 15, 24, 35, 48, 63, 80 ....`**

$$f_n=\begin{cases}0 & \text{dacă } n=1\\ 3 & \text{dacă } n=2\\ 2\cdot f_{n-1}-f_{n-2}+2 & \text{altfel}\end{cases}$$

Se citesc de la tastatură două numere naturale din intervalul **`[0,10^9]`**, **`x`** și **`y`**, reprezentând valorile a doi termeni aflați pe **poziții consecutive** în şirul dat (**`x<y`**), şi se cere să se scrie în fişierul text **`bac.txt`**, în ordine strict descrescătoare, separați prin câte un spațiu, toţi termenii şirului mai mici sau egali cu **`y`**.
Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al memoriei utilizate.
**Exemplu:** dacă se citesc numerele
**`48 63`**
fişierul **`bac.txt`** conţine numerele
**`63 48 35 24 15 8 3 0`**
**a)** Descrieţi în limbaj natural algoritmul proiectat, justificând eficienţa acestuia. **(2p.)**
**b)** Scrieţi programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 33. Informatica — Subiectul III, exercițiul 1

**Sursă: 2018 · Vara Iunie · V02**

*2018_Informatica_Mate-Info_Sesiune_Vara_Iunie-Iulie_C-Cpp_Subiect_V02.pdf*

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

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

**1.** O companie organizează cursuri de programare în limbaje din mulțimea {**PHP**, **Java**, **Python**, **C#**, **SQL**}, astfel încât o persoană poate opta pentru un curs în care se studiază un număr **par** de limbaje, dar nu poate alege Java și Python în același curs. Utilizând metoda backtracking se generează toate posibilitățile unei persoane de a opta pentru un curs în cadrul ofertei companiei. Două cursuri sunt distincte dacă diferă prin cel puțin un limbaj sau prin ordinea în care se studiază limbajele. Primele cinci soluții generate sunt, în această ordine: **(PHP, Java)**, **(PHP, Java, C#, SQL)**, **(PHP, Java, SQL, C#)**, **(PHP, Python)**, **(PHP, Python, C#, SQL)**.
Soluția generată imediat după **(Java, PHP, SQL, C#)** este: **(4p.)**

**a.** **(Java, C#)** **b.** **(Java, PHP, C#, SQL)**

**c.** **(SQL, Python)** **d.** **(SQL, Java, C#, PHP)**

**Scrieţi pe foaia de examen răspunsul pentru fiecare dintre cerinţele următoare.**

**2.** Subprogramul **f** este definit alăturat. Scrieți trei valori naturale pe care le poate avea variabila întreagă **x** astfel încât, în urma apelului de mai jos, pentru fiecare dintre acestea, să **NU** se afișeze niciun caracter *****
`f('e',x);`
**(6p.)**

```
void f(char ch, int x)
{ cout<<ch; | printf("c",ch);
  if(x==0)
    cout<<'*'; | printf("*");
  else if(ch=='a')
         cout<<x; | printf("%d",x);
       else
         f(ch-1,x-1);
}
```

**3.** Subprogramul **resturi** are patru parametri, **n**, **x**, **y** și **r**, prin care primeşte câte un număr natural din intervalul **[1,10⁹]**, **r<x<y<n**. Subprogramul returnează numărul de valori naturale din intervalul **[1,n]** pentru care atât restul împărțirii la **x**, cât și restul împărțirii la **y**, sunt egale cu **r**.
Scrieţi definiţia completă a subprogramului.
**Exemplu:** pentru **n=200**, **x=5**, **y=14** și **r=2**, subprogramul returnează numărul **3** (pentru numerele **2**, **72** și **142** atât restul împărțirii la **5**, cât și restul împărțirii la **14**, este **2**). **(10p.)**

**4.** Numim **secvenţă neuniformă** a unui șir de numere naturale un subșir al acestuia, format din termeni aflați pe poziții consecutive în șirul dat, cu proprietatea că oricare trei termeni aflați pe poziții consecutive sunt diferiți. Lungimea secvenței este egală cu numărul de termeni ai acesteia.
Fișierul **bac.txt** conține un șir de cel mult **10⁶** numere naturale din intervalul [**0**,**9**]. Numerele sunt separate prin câte un spațiu, iar în șir există cel puțin trei termeni diferiți pe poziții consecutive.
Se cere să se afișeze pe ecran lungimea maximă a unei secvențe neuniforme a șirului aflat în fișier. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al memoriei utilizate.
**Exemplu:** dacă fişierul **bac.txt** conţine numerele
**7 7 1 3 7 7 5 3 3 3 7 8 9**
atunci pe ecran se afișează valoarea **4**
**a)** Descrieţi în limbaj natural algoritmul proiectat, justificând eficienţa acestuia. **(2p.)**
**b)** Scrieţi programul C/C++ corespunzător algoritmului descris. **(8p.)**

## 34. Informatica — Subiectul III, exercițiul 1

**Sursă: 2019 · Model Oficial**

*2019_Informatica_Mate-Info_Model_Oficial_C-Cpp_Subiect.pdf*

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

**1.** Subprogramul **CifrePrime** are un singur parametru, **n**, prin care primeşte un număr natural (**n**$\in$**[0,10$^9$]**). Subprogramul returnează suma cifrelor prime ale lui **n**.
Scrieţi definiţia completă a subprogramului.
**Exemplu:** dacă **n=1235405**, atunci subprogramul returnează **15**, iar dacă **n=140**, atunci subprogramul returnează **0**. **(10p.)**

**2.** Într-un text cu cel mult **100** de caractere, cuvintele sunt formate numai din litere mici ale alfabetului englez și sunt separate prin unul sau mai multe spații.
Scrieți un program C/C++ care citește de la tastatură un astfel de text, cu cel puțin trei cuvinte, și construiește în memorie un șir de caractere format din prima consoană a primului cuvânt, urmată de prima vocală a celui de al doilea cuvânt, respectiv de ultima literă a ultimului cuvânt, în ordinea în care acestea apar în text. Șirul obținut se afișează pe ecran, iar dacă nu se poate obține un astfel de șir, se afișează pe ecran mesajul **nu exista**. Se consideră vocale literele **a**, **e**, **i**, **o**, **u**.
**Exemplu:** pentru textul **el prefera sa mearga la schi**
se afișează pe ecran șirul **lei**
iar pentru textul **ei prefera sa mearga la schi**
se afișează pe ecran mesajul **nu exista** **(10p.)**

**3.** Un interval este numit **prieten de grad n** al unui șir dacă sunt exact **n** termeni ai șirului cu valori din interval și dacă toate numerele întregi care aparțin intervalului sunt valori ale unor termeni ai șirului.
Fișierul **bac.txt** conține un șir de cel mult **10$^6$** numere naturale din intervalul **[0,10$^2$]**, separate prin câte un spațiu. Se cere să se afișeze pe ecran numărul maxim **n** cu proprietatea că există un interval prieten de grad **n** al șirului aflat în fișier. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.
**Exemplu:** dacă fișierul conține numerele
**10 10 11 3 4 2 49 4 2 3 21 2 27 12 13 14 15 5**
atunci se afișează pe ecran **8** (intervalului **[2,5]** îi aparțin **8** termeni ai șirului)
**a)** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b)** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 35. Informatica — Subiectul III, exercițiul 1

**Sursă: 2019 · Simulare Nationala**

*2019_Informatica_Mate-Info_Simulare_Nationala_C-Cpp_Subiect.pdf*

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

**1.** Subprogramul **inserare** are un singur parametru, **n**, prin care primeşte un număr natural ($n \in [10, 10^5)$). Subprogramul furnizează prin același parametru numărul obținut din **n** prin inserarea, între oricare două cifre alăturate ale sale, a valorii absolute a diferenței acestora. Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **n=7255**, atunci, după apel, **n=75235205**. **(10p.)**

**2.** Într-un text de cel mult **50** de caractere cuvintele sunt separate prin câte un spațiu și sunt formate din litere mari ale alfabetului englez, urmate eventual de caracterul **.** (punct), dacă sunt scrise prescurtat. Textul reprezintă numele unei instituții de învățământ și doar cuvintele din mulțimea {**COLEGIUL**, **LICEUL**, **NATIONAL**, **TEORETIC**} pot fi prescurtate, eliminându-se ultimele lor litere. Scrieţi un program C/C++ care citeşte de la tastatură un text de tipul precizat și construiește în memorie, apoi afișează pe ecran, numele instituției scris fără prescurtări.
**Exemplu:** dacă se citește textul **COLEG. NATIONAL DE INFORMATICA** sau textul **COLEG. NAT. DE INFORMATICA** se obține **COLEGIUL NATIONAL DE INFORMATICA** **(10p.)**

**3.** Un șir format din **2·n** numere naturale se numește **paritar** dacă fiecare dintre primii săi **n** termeni fie are aceeași paritate cu oricare dintre ultimii săi **n** termeni, fie este strict mai mic decât oricare număr de paritate diferită aflat printre aceștia.
Fișierul **bac.txt** conține numere naturale din intervalul $[0, 10^6]$: pe prima linie un număr nenul, **n**, iar pe a doua linie un șir de **2·n** numere, separate prin câte un spațiu. Se cere să se afișeze pe ecran mesajul **DA**, în cazul în care șirul aflat în fișier este paritar, sau mesajul **NU**, în caz contrar. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al memoriei utilizate.
**Exemplu:** dacă fișierul are unul dintre conținuturile de mai jos, se afișează pe ecran mesajul **DA**.

| | |
|---|---|
| 5 | 5 |
| 20 3 11 4 15 25 49 18 53 16 | 20 3 11 4 15 25 49 81 53 61 |

**a)** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b)** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 36. Informatica — Subiectul III, exercițiul 1

**Sursă: 2019 · Speciala C · V01**

*2019_Informatica_Mate-Info_Sesiune_Speciala_C-Cpp_Subiect_V01.pdf*

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

**1.** Subprogramul **Egal** are un parametru, **n**, prin care primeşte un număr natural cu cel puțin o cifră impară ($n \in [10, 10^9]$). Subprogramul returnează valoarea **1** dacă toate cifrele impare ale lui **n** sunt egale între ele sau valoarea **0** în caz contrar. Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **n=7727470** sau **n=7240** atunci subprogramul returnează **1**, iar dacă **n=7921470** atunci subprogramul returnează **0**. **(10p.)**

**2.** Într-un text cu cel mult **100** de caractere cuvintele sunt formate numai din litere mari și mici ale alfabetului englez și sunt separate prin câte un spațiu. Scrieţi un program C/C++ care citeşte de la tastatură un text de tipul precizat, apoi un număr natural, **n** ($n \in [1, 10^2)$), şi afişează pe ecran, pe linii separate, cuvintele din text care au exact **n** litere. Cuvintele sunt afişate în ordinea apariţiei lor în text, iar dacă nu există niciun astfel de cuvânt, se afișează pe ecran mesajul **nu exista**.
**Exemplu:** dacă se citește textul **Fat Frumos este cel mai viteaz**
iar **n=6**, se afişează pe ecran:
**Frumos**
**viteaz** **(10p.)**

**3.** Șirul **f** este definit astfel: $f_1=x$; $f_2=y$; $f_3=z$; $f_i=f_{i-1}+f_{i-2}-f_{i-3}$, unde **x**, **y**, **z** și **i** sunt numere naturale nenule, **i>3**.
De exemplu, dacă **x=1**, **y=2** și **z=4** șirul este: **1**, **2**, **4**, **5**, **7**, **8**, **10**, **11**, **13**, **14**, **16**, **...**
Se citesc de la tastatură un număr natural, **n** ($n \in [1, 10^4]$), apoi trei numere naturale din intervalul $[1, 10^2)$, **x**, **y** și **z**, reprezentând, în această ordine, primii trei termeni ai șirului precizat mai sus. Se cere să se scrie în fișierul **bac.txt** primii **n** termeni ai șirului, separați prin câte un spațiu, în ordine inversă a apariției lor în șir. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al spațiului de memorie utilizat.
**Exemplu:** dacă **n=10**, **x=1**, **y=2** și **z=4** fișierul conține numerele: **14 13 11 10 8 7 5 4 2 1**
**a)** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b)** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 37. Informatica — Subiectul III, exercițiul 1

**Sursă: 2019 · Toamna August · V02**

*2019_Informatica_Mate-Info_Sesiune_Toamna_August-Septembrie_C-Cpp_Subiect_V02.pdf*

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

**1.** Subprogramul **MaxImp** are doi parametri, **a** și **b**, prin care primeşte câte un număr natural ($2 \le a < b \le 400$). Subprogramul returnează cel mai mare număr natural din intervalul **[a,b]** pentru care produsul divizorilor săi impari pozitivi este strict mai mare decât el însuși sau **0**, dacă nu există niciun astfel de număr. Scrieţi definiţia completă a subprogramului.
**Exemplu:** dacă **a=14** și **b=19**, atunci subprogramul returnează **18** ($1 \cdot 3 \cdot 9 = 27 > 18$). **(10p.)**

**2.** Numim **pătrat** de dimensiune **m** al unui tablou bidimensional tabloul obținut din acesta păstrând doar elementele aflate pe primele **m** linii şi pe primele **m** coloane ale sale.
Scrieţi un program C/C++ care citeşte de la tastatură un număr natural, **n** ($n \in [2,20]$), apoi elementele unui tablou bidimensional cu **n** linii şi **n** coloane, numere naturale din intervalul $[0,10^4]$.
Programul determină un pătrat de dimensiune maximă al tabloului citit, cu toate elementele egale, și afișează pe ecran valoarea acestei dimensiuni.
**Exemplu:** pentru **n=5** și tabloul alăturat, se afişează pe ecran **3**. **(10p.)**

| | | | | |
|---|---|---|---|---|
| 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 8 |
| 2 | 2 | 2 | 2 | 2 |
| 2 | 1 | 2 | 8 | 7 |
| 3 | 5 | 2 | 1 | 2 |

**3.** Fişierul **bac.txt** conţine un şir de cel puțin două și cel mult $10^6$ numere naturale din intervalul $[0,10^3]$, separate prin câte un spaţiu. Șirul are cel puțin un termen par și cel puțin un termen impar. Se cere să se afișeze pe ecran termenii șirului, separați prin câte un spaţiu, astfel încât toți cei impari să apară înaintea tuturor celor pari, şi atât subșirul format din cei impari, cât şi subșirul format din cei pari, să fie în ordine crescătoare, ca în exemplu. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.
**Exemplu:** dacă fişierul conţine numerele **12 2 3 1 2 5**
se afişează pe ecran: **1 3 5 2 2 12**
**a)** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b)** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 38. Informatica — Subiectul III, exercițiul 1

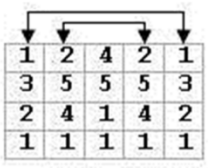
**Sursă: 2019 · Vara Iunie · V04**

*2019_Informatica_Mate-Info_Sesiune_Vara_Iunie-Iulie_C-Cpp_Subiect_V04.pdf*

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

**1.** Subprogramul **Impare** are un singur parametru, **n**, prin care primește un număr natural (**n**$\in$**[1,10$^9$]**), cu cel puțin o cifră impară. Subprogramul înlocuiește fiecare cifră impară a lui **n** cu cea mai mare cifră pară strict mai mică decât ea (astfel cifra **1** se înlocuieşte cu cifra **0**, cifra **3** cu cifra **2** etc.) și furnizează numărul obținut tot prin parametrul **n**. Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **n=235690**, atunci, după apel, **n=224680**, iar dacă **n=15690**, atunci, după apel, **n=4680**. **(10p.)**

**2.** Un tablou bidimensional cu număr impar de coloane este numit **simetric faţă de coloana din mijloc** dacă, pe fiecare linie a tabloului, elementele dispuse simetric faţă de elementul din mijloc al liniei respective au valori egale.
Scrieţi un program C/C++ care citeşte de la tastatură două numere naturale din intervalul **[3,21]**, **m** şi **n** (**n** impar), şi elementele unui tablou bidimensional cu **m** linii şi **n** coloane, numere naturale din intervalul **[0,10$^4$]**. Programul afişează pe ecran mesajul **DA**, dacă tabloul este simetric faţă de coloana din mijloc, sau mesajul **NU** în caz contrar.
**Exemplu:** pentru **m=4**, **n=5** și tabloul alăturat, se afişează pe ecran **DA** **(10p.)**




**3.** Un termen al unui șir de numere se numește **vârf local** al acestuia dacă nu există niciun alt termen mai mare sau egal cu el care să îl preceadă în șir sau dacă este egal cu termenul vecin anterior, iar acesta este vârf local.
Fișierul **bac.txt** conține un șir format din cel puțin două și cel mult **10$^6$** numere naturale din intervalul **[0,10$^3$]**, separate prin câte un spațiu. Se cere să se afișeze pe ecran, separate prin câte un spațiu, toate vârfurile locale ale șirului aflat în fișier. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al spațiului de memorie utilizat.
**Exemplu:** dacă fișierul conține numerele **7 4 9 10 10 10 8 10 10 8 30**
se afișează pe ecran **7 9 10 10 10 30**
**a)** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b)** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 39. Informatica — Subiectul III, exercițiul 1

**Sursă: 2020 · Model Oficial**

*2020_Informatica_Mate-Info_Model_Oficial_C-Cpp_Subiect.pdf*

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

**1.** Subprogramul **duplicare** are doi parametri:
- **n**, prin care primește un număr natural (**n∈[1,10⁴]**);
- **d**, prin care furnizează numărul obținut prin duplicarea fiecărei cifre impare a lui **n** sau **-1** dacă acesta nu are nicio cifră impară.

Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **n=2019**, după apel **d=201199**. **(10p.)**

**2.** Un text are cel mult **100** de caractere, iar cuvintele sale sunt formate numai din litere mici ale alfabetului englez și sunt separate prin câte un spațiu. Scrieți un program C/C++ care citește de la tastatură un număr natural **n** (**n∈[1,10²]**), apoi un text de tipul precizat mai sus, și afișează pe ecran cuvintele acestuia, pe rânduri separate, astfel încât primele poziții să fie ocupate de mulțimea formată de cele care au cel puțin **n** litere, iar următoarele poziții, în continuarea acestora, să fie ocupate de mulțimea celorlalte cuvinte.

Cuvintele din aceeași mulțime sunt afișate într-o ordine oarecare, iar dacă una dintre cele două mulțimi este vidă, se afișează pe ecran doar mesajul **nu exista**.
**Exemplu**: pentru **n=5** și textul **el mergea tot spre aleea pietruita** datele afișate pot fi cele alăturate. **(10p.)**

```
mergea
aleea
pietruita
el
tot
spre
```

**3.** Fișierul **numere.in** conține pe prima linie un număr natural **n** (**n∈[2,10⁹]**), iar pe a doua linie un șir de cel mult **10⁹** numere naturale din intervalul **[1,n]**. Numerele din șir sunt ordonate descrescător și sunt separate prin câte un spațiu. Se cere să se determine numărul valorilor naturale distincte din intervalul **[1,n]** care **NU** se găsesc în șirul menționat mai sus. Numărul determinat se afișează pe ecran. Proiectați un algoritm eficient din punctul de vedere al spațiului de memorie și al timpului de executare.
**Exemplu:** dacă fișierul conține numerele
**10**
**8 8 8 5 3 3**
se afișează pe ecran **7** (în șir nu se găsesc valorile **10 9 7 6 4 2 1**).

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

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

## 40. Informatica — Subiectul III, exercițiul 1

**Sursă: 2020 · Speciala C · V02**

*2020_Informatica_Mate-Info_Sesiune_Speciala_C-Cpp_Subiect_V02.pdf*

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

**1.** Subprogramul **multiplu** are un singur parametru, **n**, prin care primește un număr natural ($n \in [1, 10^4]$). Subprogramul returnează cel mai mic multiplu nenul al lui **n** cu proprietatea că este pătrat perfect. Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **n=72** sau **n=144**, subprogramul returnează numărul **144** ($144=12^2$). **(10p.)**

**2.** Numim **citat** într-un text o secvență de caractere din acel text care începe cu un caracter **<** și se termină cu un caracter **>**, celelalte caractere ale secvenței fiind diferite de **<** și **>**.
Un text de cel mult **100** de caractere (litere mici ale alfabetului englez, spații și caracterele **<** şi **>**) conține cel puțin un citat. Textul nu conține alte caractere **<** şi **>** decât cele care mărginesc citatele, și oricare două citate nu au nici caractere **<** şi **>** și nici alte caractere în comun.
Scrieți un program C/C++ care citește de la tastatură un text de tipul precizat și îl transformă în memorie prin înlocuirea tuturor literelor mici cuprinse în citate cu literele mari corespunzătoare, celelalte rămânând nemodificate, ca în exemplu. Programul afișează pe ecran textul obținut.
**Exemplu**: pentru textul **mai bine sa fii un <om de valoare> decat un <om de succes>**
se afișează **mai bine sa fii un <OM DE VALOARE> decat un <OM DE SUCCES>** **(10p.)**

**3.** Se numește **vârf** într-un șir de numere naturale un termen al șirului care este strict mai mare decât fiecare dintre cei doi termeni vecini cu el, aflați în șir pe poziția din stânga, respectiv din dreapta sa.
Fișierul **bac.in** conține un șir de cel mult $10^6$ numere naturale din intervalul $[0, 10^9]$, separate prin câte un spațiu. Se cere să se afișeze pe ecran vârful din șirul aflat în fișier pentru care valoarea absolută a diferenței dintre cei doi vecini ai săi este minimă. Dacă există mai multe astfel de numere, se afișează cel mai mare dintre ele, iar dacă nu există niciun vârf, se afișează pe ecran mesajul **nu exista**. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al spațiului de memorie utilizat.
**Exemplu:** dacă fișierul conține șirul **2 7 10 5 6 2 1 3 20 17 9 11 7 3 10 6 2**
se afișează pe ecran **11**
**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 41. Informatica — Subiectul III, exercițiul 1

**Sursă: 2020 · Teste Antrenament Oficiale · Test01**

*2020_Informatica_Mate-Info_Teste_Antrenament_Oficiale_C-Cpp_Subiect_Test01.pdf*

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

**1.** Subprogramul **putere** are trei parametri:

- **n**, prin care primește un număr natural din intervalul **[2,10⁹]**;
- **d** și **p**, prin care furnizează divizorul prim, **d**, care apare la cea mai mare putere, **p**, în descompunerea în factori primi a lui **n**; dacă există mai mulți astfel de divizori se afișează cel mai mare dintre ei.

Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **n=10780**, atunci, în urma apelului, **d=7** şi **p=2** (**10780=2²·5·7²·11**). **(10p.)**

**2.** Scrieţi un program C/C++ care citeşte de la tastatură două numere naturale din intervalul **[2,20]**, **n** și **k**, şi construieşte în memorie un tablou bidimensional cu **n** linii şi **n·k** coloane, numerotate începând cu **1**, astfel încât fiecare linie **i** (**i∈[1,n]**) memorează un şir crescător de termeni cu proprietatea că primul termen este **i**, fiecare valoare apare în şir de exact **k** ori și oricare doi termeni alăturați au valori egale sau consecutive.

Programul afişează pe ecran tabloul construit, fiecare linie a tabloului pe câte o linie a ecranului, cu valorile aflate pe aceeaşi linie separate prin câte un spaţiu.
**Exemplu:** dacă **n=4** și **k=3**, se afişează pe ecran tabloul alăturat.
**(10p.)**

```
1 1 1 2 2 2 3 3 3 4 4 4
2 2 2 3 3 3 4 4 4 5 5 5
3 3 3 4 4 4 5 5 5 6 6 6
4 4 4 5 5 5 6 6 6 7 7 7
```

**3.** Se consideră șirul **1**, **1**, **2**, **5**, **13**, **34**, **89**, **233**, **610** ....
definit astfel: $f_1=f_2=1$, $f_n=3\cdot f_{n-1}-f_{n-2}$ (unde **n** este un număr natural **n≥3**):
Se citesc de la tastatură două numere naturale **x** și **y** (**x≤y≤10⁹**), valorile a doi termeni aflați pe **poziții consecutive** în şirul dat, şi se cere să se scrie în fişierul text **bac.txt**, în ordine descrescătoare, separați prin câte un spațiu, toţi termenii şirului care sunt mai mici sau egali cu **y**. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate şi al timpului de executare.
**Exemplu:** dacă se citesc numerele **89 233**
fişierul **bac.txt** conţine numerele **233 89 34 13 5 2 1 1**
**a.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**
**b.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**

## 42. Informatica — Subiectul III, exercițiul 1

**Sursă: 2020 · Teste Antrenament Oficiale · Test02**

*2020_Informatica_Mate-Info_Teste_Antrenament_Oficiale_C-Cpp_Subiect_Test02.pdf*

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

**1.** Două numere distincte **a** și **b** sunt numite **d-fii** ai unui număr natural **n** dacă **a·b=n**.
Subprogramul **fii** are un singur parametru, **n**, prin care primește un număr natural (**n∈[2,10⁹]**). Subprogramul afișează pe ecran toate perechile distincte de numere naturale cu proprietatea că sunt d-fii ai lui **n**. Fiecare pereche este afișată încadrată între paranteze rotunde, numerele din pereche fiind afișate în ordine strict descrescătoare, separate printr-un spațiu. Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **n=12** se afișează pe ecran, nu neapărat în această ordine, **(12 1)(6 2)(4 3)**
iar dacă **n=16** se afișează pe ecran **(16 1)(8 2)** **(10p.)**

**2.** Un text are cel mult **100** de caractere și este format din cuvinte şi numere, separate prin câte un spațiu. Cuvintele sunt formate numai din litere ale alfabetului englez. Toate numerele sunt reale şi sunt formate numai din parte întreagă sau din parte întreagă şi parte fracţionară, separate prin virgulă (,), numerele negative fiind precedate de semnul minus (-).
Scrieţi un program C/C++ care citeşte de la tastatură textul, pe care îl transformă, eliminând din componenţa sa toate numerele negative. Programul afişează apoi pe ecran textul obţinut.
**Exemplu:** pentru textul
**2,7 minus 3,5 minus 2 egal 2,7 plus -3,5 plus -2 egal -0,2 rezultat**
se va afişa pe ecran textul:
**2,7 minus 3,5 minus 2 egal 2,7 plus plus egal rezultat** **(10p.)**

**3.** Fişierul **bac.in** conţine un şir de numere naturale distincte, din intervalul **[1,10⁹]**. Numerele din şir sunt separate prin câte un spaţiu şi cel puţin trei dintre ele au penultima cifră **2** și ultima cifră **0**.
Se cere să se afișeze pe ecran cele mai mari trei numere din şir cu proprietatea că au penultima cifră **2** și ultima cifră **0**. Numerele determinate sunt afişate în ordine crescătoare, separate prin câte un spaţiu. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate şi al timpului de executare.
**Exemplu**: dacă fişierul conţine numerele **9731 50 112 20 8 16 8520 3 2520 1520**
pe ecran se vor afişa, în această ordine, numerele: **1520 2520 8520**
**a.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**
**b.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**

## 43. Informatica — Subiectul III, exercițiul 1

**Sursă: 2020 · Teste Antrenament Oficiale · Test03**

*2020_Informatica_Mate-Info_Teste_Antrenament_Oficiale_C-Cpp_Subiect_Test03.pdf*

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

**1.** Subprogramul **factori** are doi parametri, **n** şi **m**, prin care primește câte un număr natural din intervalul $[1,10^9]$. Subprogramul returnează numărul valorilor prime care se regăsesc atât în descompunerea în factori primi a lui **n**, cât și în descompunerea în factori primi a lui **m**.
Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **n=750** şi **m=490**, atunci subprogramul returnează **2** ($750=\underline{2}\cdot 3\cdot \underline{5}^3$, $490=\underline{2}\cdot\underline{5}\cdot 7^2$). **(10p.)**

**2.** Scrieţi un program **C/C++** care citeşte de la tastatură un număr natural, **n** ($n\in[2,20]$), şi construieşte în memorie un tablou bidimensional cu **n** linii şi **n** coloane, având proprietăţile:
- toate elementele situate pe diagonala secundară sunt nule;
- fiecare linie conţine, începând cu diagonala secundară, de la dreapta la stânga, un şir strict crescător de numere consecutive, iar începând cu diagonala secundară, de la stânga la dreapta, tot un şir strict crescător de numere consecutive.

Programul afişează pe ecran tabloul construit, fiecare linie a tabloului pe câte o linie a ecranului, cu elementele aflate pe aceeaşi linie separate prin câte un spaţiu.
**Exemplu:** dacă **n=5** se afişează pe ecran tabloul alăturat. **(10p.)**

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

**3.** Fişierul **bac.in** conţine un şir de cel mult $10^6$ numere întregi din intervalul $[-10^9,10^9]$, separate prin câte un spaţiu. Cel puţin un număr din șir este negativ.
Se cere să se afişeze pe ecran lungimea maximă a unei secvenţe a şirului care fie începe, fie se încheie cu un număr negativ. O secvenţă este formată din termeni aflaţi pe poziţii consecutive în şir, iar lungimea secvenței este egală cu numărul de termeni ai acesteia. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate şi al timpului de executare.
**Exemplu:** dacă fişierul conţine numerele **12 25 -6 7 80 -75 101 -6 52 -124 87 99 210**
pe ecran se afişează **11** (corespunzător secvenţei **-6 7 80 -75 101 -6 52 -124 87 99 210**).
**a.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**
**b.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**

## 44. Informatica — Subiectul III, exercițiul 1

**Sursă: 2020 · Teste Antrenament Oficiale · Test04**

*2020_Informatica_Mate-Info_Teste_Antrenament_Oficiale_C-Cpp_Subiect_Test04.pdf*

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

**1.** Două numere **a** și **b** sunt numite **generatoare** ale unui număr natural **n** dacă **a·b+[a/b]=n**, unde s-a notat cu **[c]** partea întreagă a numărului real **c**.
Subprogramul **generatoare** are un singur parametru, **n**, prin care primește un număr natural (**n**∈**[2,10⁹]**). Subprogramul afișează pe ecran toate perechile distincte de numere naturale cu proprietatea că sunt generatoare ale lui **n** și că primul număr din pereche este par. Numerele din fiecare pereche sunt separate prin simbolul minus (-), iar perechile sunt separate prin câte un spațiu. Dacă nu există astfel de perechi, se afișează pe ecran mesajul **nu exista**. Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **n=2020** se afișează pe ecran
**2-1010 4-505 10-202 20-101 96-21 200-10 606-3 808-2 1010-1** **(10p.)**

**2.** Într-un tablou bidimensional, cu elemente având valori numai în mulţimea **{0,1}**, numim coloane „complementare” două coloane cu proprietatea că oricare două elemente ale acestora, aflate pe aceeaşi linie, sunt diferite.
Scrieţi un program C/C++ care citeşte de la tastatură două numere naturale din intervalul **[2,20]**, **m** şi **n**, şi elementele unui tablou bidimensional cu **m** linii şi **n** coloane, numere naturale din mulţimea **{0,1}**. Programul afişează pe ecran numărul de coloane ale tabloului care sunt „complementare" cu prima coloană a acestuia.
**Exemplu:** dacă **m=3**, **n=6**, pentru tabloul alăturat se afişează pe ecran **3**. **(10p.)**

| | | | | | |
|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 |

**3.** Fişierul **bac.txt** conţine, **în ordine descrescătoare**, cel puţin două şi cel mult **10⁶** numere naturale din intervalul **[0,10⁹]**, separate prin câte un spaţiu. Se cere să se afişeze pe ecran, în ordine strict descrescătoare, separate prin câte un spaţiu, numai numerele care apar în fişier de exact două ori. Dacă nu există niciun astfel de număr, se afișează pe ecran mesajul **nu exista**. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate şi al timpului de executare.
**Exemplu:** dacă fişierul conţine numerele **100 50 50 50 49 49 36 16 16 12 10 10 9 7 7**
pe ecran se afişează, în această ordine, numerele **49 16 10 7**
**a.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**
**b.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**

## 45. Informatica — Subiectul III, exercițiul 1

**Sursă: 2020 · Teste Antrenament Oficiale · Test05**

*2020_Informatica_Mate-Info_Teste_Antrenament_Oficiale_C-Cpp_Subiect_Test05.pdf*

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

**1.** Un număr este scris în baza de numerație **b** (**b≤10**) dacă cifrele sale aparțin intervalului **[0,b-1]**.
Subprogramul **baza** are un singur parametru, **n**, prin care primește un număr natural (**n∈[0,10⁹]**). Subprogramul returnează cea mai mică bază din intervalul **[2,10]** căreia i-ar putea corespunde scrierea lui **n**. Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **n=50731**, subprogramul returnează numărul **8**. **(10p.)**

**2.** Un text cu cel mult **100** de caractere conține cuvinte și numere, separate prin câte un spațiu. Cuvintele sunt formate numai din litere mici ale alfabetului englez, iar numerele sunt reale, pozitive, cu partea zecimală și partea întreagă separate prin simbolul virgulă, sau numai cu partea întreagă, ca în exemplu. Scrieți un program C/C++ care citește de la tastatură un text de tipul precizat și afișează pe ecran numărul de valori întregi din text.
**Exemplu:** pentru textul
**grus leucogeranus are 1,40 m inaltime si traieste intre 30 si 40 de ani**
se afișează pe ecran **2** **(10p.)**

**3.** Fișierul **bac.txt** conține un șir de cel mult **10⁶** numere întregi din intervalul **[-10³,10³]**, separate prin câte un spațiu. Se cere să se afișeze pe ecran suma maximă obținută adunând numere de pe poziții consecutive în șirul aflat în fișier. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.
**Exemplu:** dacă fișierul **bac.txt** conține valorile **4 -6 7 2 -1 4 -10 -3 9 2 -2**
se afișează pe ecran numărul **12**

**a.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**
**b.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**

## 46. Informatica — Subiectul III, exercițiul 1

**Sursă: 2020 · Teste Antrenament Oficiale · Test06**

*2020_Informatica_Mate-Info_Teste_Antrenament_Oficiale_C-Cpp_Subiect_Test06.pdf*

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

**1.** Subprogramul **prodprim** are doi parametri:
- **n**, prin care primește un număr natural ($n \in [2, 10^9]$);
- **p**, prin care furnizează produsul divizorilor primi ai lui **n**.

Scrieți definiția completă a subprogramului.
**Exemplu**: dacă $n=2000$, în urma apelului $p=10$, deoarece $2000=2^4 \cdot 5^3$. **(10p.)**

**2.** Într-un text cu cel mult **100** de caractere, cuvintele sunt formate din litere mici ale alfabetului englez și sunt separate prin câte un spațiu. Scrieți un program C/C++ care citește de la tastatură un text de tipul menționat și afișează pe ecran, pe linii separate, toate cuvintele sale pentru care numărul de vocale este strict mai mic decât numărul de consoane. Dacă nu există niciun astfel de cuvânt, se afișează pe ecran mesajul **nu exista**. Se consideră vocale literele din mulțimea **a**, **e**, **i**, **o**, **u**.

**Exemplu**: pentru textul **ei au plantat tamarix ea a adus iasomie** se afișează pe ecran, nu neapărat în această ordine, cuvintele alăturate. **(10p.)**

```
plantat
tamarix
```

**3.** Se citesc de la tastatură două numere naturale din intervalul **[1,81]**, **p1** și **p2**, și se cere scrierea în fișierul **bac.out** a tuturor numerelor naturale cu exact **7** cifre, pentru care produsul primelor două cifre este egal cu **p1**, cele trei cifre din mijloc sunt egale între ele, iar produsul ultimelor două cifre este egal cu **p2**. Numerele apar în fișier în ordine strict descrescătoare, fiecare pe câte o linie. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.
**Exemplu:** dacă $p1=12$, iar $p2=8$, atunci **2633324** și **3400018** sunt două dintre cele **160** de numere cu proprietatea cerută ($2 \cdot 6 = 3 \cdot 4 = 12$ și $2 \cdot 4 = 1 \cdot 8 = 8$).

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

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

## 47. Informatica — Subiectul III, exercițiul 1

**Sursă: 2020 · Teste Antrenament Oficiale · Test07**

*2020_Informatica_Mate-Info_Teste_Antrenament_Oficiale_C-Cpp_Subiect_Test07.pdf*

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

**1.** Subprogramul **putere** are doi parametri, **n** și **p**, prin care primește câte un număr natural ($n \in [2, 10^9]$, $p \in [0, 10^9]$). Subprogramul returnează puterea la care apare numărul **p** în descompunerea în factori primi a lui **n**, dacă **p** este număr prim, sau valoarea **-1** în caz contrar.
Scrieți definiția completă a subprogramului.
**Exemplu**: dacă **n=80** și **p=2**, subprogramul returnează numărul **4** ($80 = 2^4 \cdot 5$). **(10p.)**

**2.** O valoare **k polarizează** două șiruri dacă există doi termeni care au acea valoare, unul fiind în primul șir, iar celălalt în al doilea șir.
Scrieți un program C/C++ care citește de la tastatură numere naturale din intervalul **[1,20]**: **m**, **n** și elementele unui tablou bidimensional cu **m** linii și **n** coloane, cu proprietatea că nu există două elemente egale situate pe aceeași linie sau pe aceeași coloană. Programul afișează pe ecran valorile care pot polariza două șiruri, și anume șirul format din elementele de pe prima coloană, respectiv șirul format din elementele ultimei coloane a tabloului.
Valorile sunt afișate într-o ordine oarecare, separate prin câte un spațiu, iar dacă nu există astfel de valori se afișează pe ecran mesajul **nepolarizate**.
**Exemplu:** pentru **m=4**, **n=5** și tabloul alăturat se afișează pe ecran, nu neapărat în această ordine, numerele **5 6** **(10p.)**

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

**3.** Fișierul **bac.txt** conține un șir de cel mult $10^6$ numere întregi din intervalul $[-10^3, 10^3]$, separate prin câte un spațiu. Se cere să se afișeze pe ecran suma minimă obținută adunând numere de pe poziții consecutive în șirul aflat în fișier. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.
**Exemplu:** dacă fișierul **bac.txt** conține valorile **-4 6 -7 -2 1 -4 10 3 -9 -2 2**
se afișează pe ecran numărul **-12**

**a.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**
**b.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**

## 48. Informatica — Subiectul III, exercițiul 1

**Sursă: 2020 · Teste Antrenament Oficiale · Test08**

*2020_Informatica_Mate-Info_Teste_Antrenament_Oficiale_C-Cpp_Subiect_Test08.pdf*

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

**1.** Subprogramul **suma** are un singur parametru, **n**, prin care primește un număr natural ($n \in [2, 10^9]$). Subprogramul returnează suma divizorilor primi ai lui **n**. Scrieți definiția completă a subprogramului.
**Exemplu:** pentru **n=12** subprogramul returnează **5** (divizorii primi ai lui **12** sunt **2** și **3**). **(10p.)**

**2.** Scrieți un program C/C++ care citește de la tastatură numere naturale: **n** ($n \in [2,20]$), **k** ($k \in [1,n]$) și $n \cdot n$ numere din intervalul $[0,10^9]$, elemente ale unui tablou bidimensional cu **n** linii și **n** coloane, numerotate începând de la **1**. Programul transformă tabloul în memorie, deplasând circular spre stânga, cu câte o poziție, toate elementele situate pe linia **k**, în stânga diagonalei principale, ca în exemplu. Elementele tabloului obținut sunt afișate pe ecran, fiecare linie pe câte o linie a ecranului, cu elementele fiecărei linii separate prin câte un spațiu.

**Exemplu:** pentru **n=6**, **k=4** și tabloul

| | | | | | |
|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 |
| 6 | 7 | 8 | 9 | 0 | 1 |
| 2 | 4 | 6 | 8 | 0 | 2 |
| 1 | 3 | 5 | 7 | 9 | 5 |
| 6 | 2 | 7 | 4 | 5 | 7 |
| 3 | 9 | 4 | 8 | 5 | 7 |

se obține tabloul

| | | | | | |
|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 |
| 6 | 7 | 8 | 9 | 0 | 1 |
| 2 | 4 | 6 | 8 | 0 | 2 |
| 3 | 5 | 1 | 7 | 9 | 5 |
| 6 | 2 | 7 | 4 | 5 | 7 |
| 3 | 9 | 4 | 8 | 5 | 7 |

**(10p.)**

**3.** Fișierul **bac.in** conține un șir de cel mult $10^6$ numere naturale din intervalul $[0,10^9]$, separate prin câte un spațiu. Se cere să se afișeze pe ecran pozițiile din șir pe care se află termeni precedați de un număr maxim de valori care au cifra unităților egală cu cifra unităților lor. Numerele afișate sunt separate prin câte un spațiu. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.
**Exemplu:** dacă fișierul **bac.in** conține numerele **112 12 5 25 88 15 2 19 32 179 35 621**
pe ecran se afișează numerele de mai jos (termenii **32**, respectiv **35** respectă proprietatea cerută):
**9 11**

**a.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**
**b.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**

## 49. Informatica — Subiectul III, exercițiul 1

**Sursă: 2020 · Teste Antrenament Oficiale · Test09**

*2020_Informatica_Mate-Info_Teste_Antrenament_Oficiale_C-Cpp_Subiect_Test09.pdf*

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

**1.** Subprogramul **suma** are doi parametri:
- **n**, prin care primește un număr natural din intervalul **[0,10⁹]**;
- **s**, prin care furnizează suma cifrelor impare distincte din scrierea acestuia.

Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **n=4713835**, după apel **s=16** (**16=7+1+3+5**), iar dacă **n=48**, după apel **s=0**. **(10p.)**

**2.** Un cuvânt este **prefix** al unui alt cuvânt dacă se obține din acesta, prin eliminarea ultimelor sale litere. Scrieți un program C/C++ care citește de la tastatură un număr natural **n** (**n∈[2,20]**) și apoi **n** cuvinte distincte, fiecare fiind format din cel mult **20** de caractere, numai litere mici ale alfabetului englez.

La introducerea datelor, după fiecare cuvânt se tastează Enter. Programul afișează pe ecran, separate prin câte un spațiu, cuvintele care îl au drept prefix pe ultimul cuvânt citit. Dacă nu există astfel de cuvinte, se afișează pe ecran mesajul **nu exista**.
**Exemplu:** dacă **n=6** și se citesc cuvintele alăturate, pe ecran se afișează
**raita raid raion** **(10p.)**

```
raita
grai
raid
raion
straie
rai
```

**3.** Numim **k-secvență** într-un șir de numere naturale, o succesiune de termeni aflați pe poziții consecutive în șir, cu proprietatea că sunt divizibili cu numărul natural nenul **k**. **Lungimea** secvenței este egală cu numărul de termeni ai săi.
Fișierul **bac.txt** conține numere naturale din intervalul **[0,10⁹]**: pe prima linie un număr nenul **k**, iar pe a doua linie un șir de cel mult **10⁶** numere, separate prin câte un spațiu. Cel puțin un termen din șir este divizibil cu **k**. Se cere să se afișeze pe ecran două valori, separate printr-un spațiu, reprezentând lungimea maximă a unei k-secvențe din șirul aflat în fișier, respectiv numărul de astfel de secvențe. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.

**Exemplu:** dacă fișierul are conținutul alăturat, se afișează **4 2**

```
5
2 10 5 20 21 0 10 60 15 3 9 20 20 5 45
```

**a.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**
**b.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**

## 50. Informatica — Subiectul III, exercițiul 1

**Sursă: 2020 · Teste Antrenament Oficiale · Test10**

*2020_Informatica_Mate-Info_Teste_Antrenament_Oficiale_C-Cpp_Subiect_Test10.pdf*

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

**1.** Subprogramul **produs** are doi parametri:
- **n**, prin care primește un număr natural ($n \in [0, 10^9]$);
- **p**, prin care furnizează produsul cifrelor pare distincte din scrierea acestuia, sau **-1** dacă nu există astfel de cifre.

Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **n=1622325**, după apel **p=12** (**12=6·2**), iar dacă **n=122325**, după apel **p=2**. **(10p.)**

**2.** Într-un text cu cel mult $10^2$ caractere cuvintele sunt formate din litere mici ale alfabetului englez și sunt separate prin câte un spațiu. Scrieți un program C/C++ care citește de la tastatură un text de tipul menționat, pe care îl modifică în memorie, inserând cuvântul **succes** între oricare două cuvinte ale sale care se termină cu aceeași literă. Cuvântul inserat este separat prin câte un spațiu de cuvintele vecine. Textul transformat este afișat pe ecran, iar dacă nu există perechi de astfel de cuvinte, se afișează pe ecran mesajul **nu exista**.
**Exemplu:** dacă textul citit este **testez validez utilizez date corecte acum**
se obține textul **testez succes validez succes utilizez date succes corecte acum** **(10p.)**

**3.** Un șir format din cel puțin trei termeni formează o progresie aritmetică de rație **r** dacă diferența dintre oricare termen al acestuia și cel aflat pe poziția consecutivă în șir este egală cu **r**.
Fișierul text **bac.txt** conține un șir de cel puțin trei și cel mult $10^6$ numere întregi din intervalul $[-10^8, 10^8]$. Numerele sunt separate prin câte un spațiu. Se cere să se afișeze pe ecran rația unei secvențe din șir cu număr maxim de termeni, secvență care formează o progresie aritmetică. Dacă există mai multe astfel de secvențe de lungime maximă se afișează rația cea mai mare, iar dacă nu există nicio astfel de secvență, se afișează pe ecran mesajul **nu exista**. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.
**Exemplu:** dacă fișierul conține numerele **4 9 14 19 18 16 8 5 3 1 -1 -3 -5 -7**
pe ecran se afișează valoarea **-2** (corespunzătoare secvenței **5 3 1 -1 -3 -5 -7**).
**a.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**
**b.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**

## 51. Informatica — Subiectul III, exercițiul 1

**Sursă: 2020 · Teste Antrenament Oficiale · Test11**

*2020_Informatica_Mate-Info_Teste_Antrenament_Oficiale_C-Cpp_Subiect_Test11.pdf*

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

**1.** Subprogramul **patrate** are doi parametri, **x** și **y**, prin care primește câte un număr natural ($1 \le x \le y \le 10^9$). Subprogramul afișează pe ecran o expresie aritmetică reprezentând suma numerelor din intervalul **[x,y]** care au proprietatea că sunt pătrate perfecte, urmate de valoarea acestei sume. Termenii sumei sunt într-o ordine oarecare și sunt separați prin câte un simbol plus (**+**), iar valoarea sumei este precedată de simbolul egal (**=**), ca în exemplu. Dacă nu există niciun astfel de termen, se afișează pe ecran mesajul **nu exista**.
Scrieți definiția completă a subprogramului.
**Exemplu**: dacă **x=10** și **y=50** se poate afișa pe ecran **16+25+36+49=126** **(10p.)**

**2.** Într-un text cu cel mult $10^2$ caractere, cuvintele sunt formate din litere mici și mari ale alfabetului englez și sunt separate prin câte un spațiu. Scrieți un program C/C++ care citește de la tastatură un text de tipul precizat, pe care îl transformă, astfel încât fiecare cuvânt să aibă prima literă mare, și toate celelalte litere mici. Textul obținut se afișează pe ecran.
**Exemplu:** dacă de la tastatură se introduce textul **ABIA aSTept sa Merg lA scoala**
se obține textul **Abia Astept Sa Merg La Scoala** **(10p.)**

**3.** Fişierul **bac.txt** conţine un şir **crescător** de cel mult $10^6$ numere naturale din intervalul $[0,10^9]$, separate prin câte un spaţiu. Se cere să se afişeze pe ecran fiecare număr distinct din şir, urmat de numărul de apariţii ale acestuia în şir. Numerele afișate sunt separate prin câte un spațiu. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.
**Exemplu:** dacă fişierul **bac.txt** conține numerele **0 0 0 5 5 5 5 7 7 11 20 20**
se afișează **0 3 5 4 7 2 11 1 20 2**
**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 52. Informatica — Subiectul III, exercițiul 1

**Sursă: 2020 · Teste Antrenament Oficiale · Test12**

*2020_Informatica_Mate-Info_Teste_Antrenament_Oficiale_C-Cpp_Subiect_Test12.pdf*

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

**1.** Subprogramul **pDoi** are un singur parametru, **n**, prin care primește un număr natural ($n \in [1, 10^9]$). Subprogramul returnează cea mai mare valoare din intervalul **[1,n]**, cu proprietatea că este o putere a lui **2**.
Scrieți definiția completă a subprogramului.
**Exemplu**: dacă **n=20**, subprogramul returnează **16**. **(10p.)**

**2.** Într-un text cu cel mult $10^2$ caractere, cuvintele sunt formate din litere mici ale alfabetului englez și sunt separate prin câte un spațiu. Scrieți un program C/C++ care citește de la tastatură un text de tipul precizat, pe care îl transformă în memorie, eliminând numai ultima vocală care apare în text, ca în exemplu. Programul afișează pe ecran textul obținut sau mesajul **nu exista**, dacă în text nu există nicio vocală. Se consideră vocale literele **a**, **e**, **i**, **o**, **u**.
**Exemplu**: dacă se citește textul: **cuvantul ritm poate fi tradus rhythm**
se obține textul **cuvantul ritm poate fi trads rhythm** **(10p.)**

**3.** Șirul de mai jos este definit alăturat:
**1**, **2**, **3**, **4**, **5**, **6**, **7**, **8**, **9**, **10**, **20**, **21**, **42**, **43**, **86**....

$$f_n = \begin{cases} n & \text{dacă } n \le 10 \\ 2 \cdot f_{n-1} & \text{dacă } n > 10 \text{ și } n \text{ impar} \\ 1 + f_{n-1} & \text{altfel} \end{cases}$$

Se citește de la tastatură un număr natural **x** ($x \in [1, 10^9]$), termen al șirului dat, și se cere să se scrie în fișierul **bac.out**, separați prin câte un spațiu, în ordine strict descrescătoare, toți termenii șirului mai mici sau egali cu **x**. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.

**Exemplu:** dacă valoarea citită de la tastatură este **43**

se scriu în fișier numerele **43 42 21 20 10 9 8 7 6 5 4 3 2 1**

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

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

## 53. Informatica — Subiectul III, exercițiul 1

**Sursă: 2020 · Teste Antrenament Oficiale · Test13**

*2020_Informatica_Mate-Info_Teste_Antrenament_Oficiale_C-Cpp_Subiect_Test13.pdf*

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

**1.** Subprogramul **putere** are trei parametri:

- **n**, prin care primește un număr natural din intervalul **[2,10⁹]**;
- **d** și **p**, prin care furnizează divizorul prim, **d**, care apare la cea mai mică putere, **p**, în descompunerea în factori primi a lui **n**; dacă există mai mulți astfel de divizori se afișează cel mai mic dintre ei.

Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **n=10780**, atunci, în urma apelului, **d=5** şi **p=1** (**10780=2²·5·7²·11**). **(10p.)**

**2.** Scrieţi un program C/C++ care citeşte de la tastatură două numere naturale din intervalul **[2,20]**, **n** și **k**, şi construieşte în memorie un tablou bidimensional cu **n·k** linii şi **n** coloane, numerotate începând cu **1**, astfel încât fiecare coloană **i** (**i∈[1,n]**) memorează un şir crescător de termeni cu proprietatea că primul termen este **i**, fiecare valoare apare în şir de exact **k** ori și oricare doi termeni alăturați au valori egale sau consecutive.
Programul afişează pe ecran tabloul construit, fiecare linie a tabloului pe câte o linie a ecranului, cu valorile aflate pe aceeaşi linie separate prin câte un spaţiu.
**Exemplu:** dacă **n=4** și **k=3**, se afişează pe ecran tabloul alăturat. **(10p.)**

```
1 2 3 4
1 2 3 4
1 2 3 4
2 3 4 5
2 3 4 5
2 3 4 5
3 4 5 6
3 4 5 6
3 4 5 6
4 5 6 7
4 5 6 7
4 5 6 7
```

**3.** Șirul de mai jos este definit astfel: $f_1=1$, $f_2=2$, $f_n=3\cdot f_{n-1}-2\cdot f_{n-2}$ (unde **n** este un număr natural **n≥3**).
**1**, **2**, **4**, **8**, **16**, **32**....
Se citește de la tastatură un număr natural **x** (**x≤10⁹**), valoare a unui termen al șirului dat, şi se cere să se scrie în fişierul text **bac.txt**, în ordine descrescătoare, separați prin câte un spațiu, toţi termenii şirului care sunt mai mici sau egali cu **x**. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate şi al timpului de executare.
**Exemplu:** dacă se citește numărul **16**
fişierul **bac.txt** conţine numerele **16 8 4 2 1**
**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 54. Informatica — Subiectul III, exercițiul 1

**Sursă: 2020 · Teste Antrenament Oficiale · Test14**

*2020_Informatica_Mate-Info_Teste_Antrenament_Oficiale_C-Cpp_Subiect_Test14.pdf*

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

**1.** Două numere **a** și **b** (**a<b**) sunt numite **divizori pereche** ai unui număr natural **n** dacă **a·b=n**.
Subprogramul **perechi** are un singur parametru, **n**, prin care primește un număr natural (**n∈[2,10⁹]**). Subprogramul afișează pe ecran toate perechile distincte de numere naturale cu paritate diferită cu proprietatea că sunt divizori pereche ai lui **n**. Fiecare pereche este afișată încadrată între paranteze drepte, numerele din pereche fiind afișate în ordine strict crescătoare, separate printr-un spațiu, iar dacă nu există astfel de perechi, se afișează pe ecran mesajul **nu exista**. Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **n=12** se afișează pe ecran, nu neapărat în această ordine, **[1 12][3 4]**
iar dacă **n=9** se afișează pe ecran **nu exista** **(10p.)**

**2.** Un text are cel mult **100** de caractere și este format din cuvinte și numere, separate prin câte un spațiu. Cuvintele sunt formate numai din litere ale alfabetului englez. Toate numerele sunt reale și sunt formate numai din parte întreagă sau din parte întreagă și parte fracționară, separate prin virgulă (**,**), numerele negative fiind precedate de semnul minus (**-**). Cel puțin unul dintre numerele reale este negativ. Scrieți un program C/C++ care citește de la tastatură textul, pe care îl transformă în memorie, înlocuind fiecare număr negativ cu valoarea sa absolută. Programul afișează apoi pe ecran textul obținut.
**Exemplu:** pentru textul **Modul de -3,24 este 3,24 si modul de -15 este 15**
se va afișa pe ecran textul: **Modul de 3,24 este 3,24 si modul de 15 este 15** **(10p.)**

**3.** Fișierul **bac.in** conține un șir de numere naturale distincte, din intervalul **[1,10⁹]**. Numerele din șir sunt separate prin câte un spațiu și cel puțin trei dintre ele au penultima cifră **2** și ultima cifră **0**.
Se cere să se afișeze pe ecran cele mai mici trei numere din șir cu proprietatea că au penultima cifră **2** și ultima cifră **0**. Numerele determinate sunt afișate în ordine crescătoare, separate prin câte un spațiu. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.
**Exemplu**: dacă fișierul conține numerele **9731 50 112 20 8 16 8520 3 2520 1520**
pe ecran se vor afișa, în această ordine, numerele: **20 1520 2520**
**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 55. Informatica — Subiectul III, exercițiul 1

**Sursă: 2020 · Teste Antrenament Oficiale · Test15**

*2020_Informatica_Mate-Info_Teste_Antrenament_Oficiale_C-Cpp_Subiect_Test15.pdf*

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

**1.** Subprogramul **divPrimMax** are doi parametri:
- **n**, prin care primește un număr natural ($n \in [2, 10^9]$);
- **p**, prin care furnizează cel mai mare divizor prim al lui **n**.

Scrieți definiția completă a subprogramului.
**Exemplu**: dacă $n=2000$, în urma apelului $p=5$, deoarece $2000=2^4 \cdot 5^3$. **(10p.)**

**2.** Într-un text cu cel mult **100** de caractere, cuvintele sunt formate din litere mici ale alfabetului englez și sunt separate prin câte un spațiu. Scrieți un program C/C++ care citește de la tastatură un text de tipul menționat și afișează pe ecran numărul de cuvinte ale sale formate dintr-un număr egal de vocale și consoane. Se consideră vocale literele din mulțimea **a**, **e**, **i**, **o**, **u**.

**Exemplu**: pentru textul
**cuvantul consoane are un numar de patru vocale si patru consoane**
se afișează pe ecran **6**. **(10p.)**

**3.** Se citesc de la tastatură două numere naturale din intervalul **[1,81]**, **p1** și **p2**, și se cere scrierea în fișierul **bac.out** a tuturor numerelor naturale cu exact **7** cifre, pentru care produsul primelor două cifre este egal cu **p1**, cele trei cifre din mijloc sunt egale între ele, iar produsul ultimelor două cifre este egal cu **p2**. Numerele apar în fișier în ordine strict crescătoare, fiecare pe câte o linie. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.
**Exemplu:** dacă **p1=12**, iar **p2=8**, atunci **2633324** și **3400018** sunt două dintre cele **160** de numere cu proprietatea cerută ($2 \cdot 6 = 3 \cdot 4 = 12$ și $2 \cdot 4 = 1 \cdot 8 = 8$).
**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 56. Informatica — Subiectul III, exercițiul 1

**Sursă: 2020 · Teste Antrenament Oficiale · Test16**

*2020_Informatica_Mate-Info_Teste_Antrenament_Oficiale_C-Cpp_Subiect_Test16.pdf*

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

**1.** Subprogramul **nrDivPrimi** are un singur parametru, **n**, prin care primește un număr natural ($n \in [2, 10^9]$). Subprogramul returnează numărul divizorilor care, în descompunerea în factori primi a lui **n**, apar la o putere impară.
Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **n=9000**, subprogramul returnează **2** ($9000 = \underline{2}^3 \cdot 3^2 \cdot \underline{5}^3$). **(10p.)**

**2.** Scrieți un program C/C++ care citește de la tastatură două numere naturale din intervalul $[2, 10^2]$, **n** și **m**, și construiește în memorie un tablou bidimensional cu **n** linii și **m** coloane, cu proprietatea că parcurgându-l linie cu linie de sus în jos și fiecare linie de la stânga la dreapta, se obține șirul primelor **n*m** pătrate perfecte impare, ordonat strict descrescător, ca în exemplu.

Elementele tabloului obținut se afișează pe ecran, fiecare linie a tabloului pe câte o linie a ecranului, valorile de pe aceeași linie fiind separate prin câte un spațiu.
**Exemplu:** pentru **n=2**, **m=3** se obține tabloul alăturat. **(10p.)**

```
121 81 49
 25  9  1
```

**3.** Fișierul **bac.in** conține numere naturale: pe prima linie două numere din intervalul $[1, 10^6]$, **m** și **n**, pe a doua linie un șir de m numere din intervalul $[1, 10^9]$, iar pe a treia linie un șir de **n** numere din intervalul $[1, 10^9]$. Numerele aflate pe aceeași linie a fișierului sunt separate prin câte un spațiu, și ambele șiruri sunt ordonate crescător.
Se cere să se afișeze pe ecran, în ordine strict crescătoare, un șir format dintr-un număr maxim de termeni care aparțin cel puțin unuia dintre cele două șiruri, astfel încât oricare două elemente aflate pe poziții consecutive să fie de paritate diferită. Numerele afișate sunt separate prin câte un spațiu.

Proiectați un algoritm eficient din punctul de vedere al timpului de executare.
**Exemplu**: dacă fișierul are conținutul alăturat, se afișează pe ecran
**2 3 4 5 8 11 14** sau **2 3 4 5 10 11 14**

```
8 5
2 4 5 8 8 11 14 14
3 4 5 5 10
```

**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 57. Informatica — Subiectul III, exercițiul 1

**Sursă: 2020 · Teste Antrenament Oficiale · Test17**

*2020_Informatica_Mate-Info_Teste_Antrenament_Oficiale_C-Cpp_Subiect_Test17.pdf*

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

**1.** Subprogramul **maxim** are un singur parametru, **n**, prin care primește un număr natural ($n \in [0, 10^9]$). Subprogramul returnează cea mai mare cifră impară din scrierea acestuia, sau **-1** dacă nu există astfel de cifre. Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **n=567283**, subprogramul returnează **7**. **(10p.)**

**2.** Într-un text cu cel mult $10^2$ caractere cuvintele sunt formate din litere mici ale alfabetului englez și sunt separate prin câte un spațiu. Scrieți un program C/C++ care citește de la tastatură un text de tipul menționat, pe care îl modifică în memorie, duplicând fiecare cuvânt format numai din vocale. Cuvântul duplicat este separat prin câte un spațiu de cuvintele vecine. Textul transformat este afișat pe ecran, iar dacă nu există astfel de cuvinte, se afișează pe ecran mesajul **nu exista**.
**Exemplu:** dacă textul citit este **oaia aia alba e a ei**
se obține textul **oaia oaia aia aia alba e e a a ei ei** **(10p.)**

**3.** Fișierul text **bac.txt** conține numere naturale din intervalul $[1, 10^4]$: pe prima linie un număr **n**, pe a doua linie un șir de **n** numere, iar pe fiecare dintre următoarele linii, până la finalul fișierului, câte o pereche de numere, reprezentând extremitățile unui interval închis. Numerele aflate pe aceeași linie a fișierului sunt în ordine crescătoare și sunt separate prin câte un spațiu.

Se cere să se afișeze pe ecran numărul de intervale care nu conțin niciun termen al șirului aflat pe a doua linie a fișierului. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.
**Exemplu:** dacă fișierul conține numerele alăturate, se afișează pe ecran **3**.
**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

```
5
4 8 9 16 25
1 3
2 5
9 15
5 7
20 100
10 12
```

## 58. Informatica — Subiectul III, exercițiul 1

**Sursă: 2020 · Teste Antrenament Oficiale · Test18**

*2020_Informatica_Mate-Info_Teste_Antrenament_Oficiale_C-Cpp_Subiect_Test18.pdf*

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

**1.** Subprogramul **suma** are doi parametri:
- **n**, prin care primește un număr natural din intervalul $[0,10^9]$;
- **s**, prin care furnizează suma cifrelor pare distincte din scrierea acestuia.

Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **n=67638825**, după apel **s=16** (**16=6+8+2**), iar dacă **n=15**, după apel **s=0**. **(10p.)**

**2.** Un cuvânt este **sufix** al unui alt cuvânt dacă se obține din acesta, prin eliminarea primelor sale litere.
Scrieți un program C/C++ care citește de la tastatură două numere naturale **n** și **k** (**n**$\in$**[2,20]**, **k**$\in$**[1,n]**) și apoi **n** cuvinte distincte, fiecare fiind format din cel mult **20** de caractere, numai litere mici ale alfabetului englez.

La introducerea datelor, după fiecare cuvânt se tastează Enter. Programul afișează pe ecran, separate prin câte un spațiu, cuvintele care îl au drept sufix pe al **k**-lea cuvânt citit, ca în exemplu. Dacă nu există astfel de cuvinte, se afișează pe ecran mesajul **nu exista**.
**Exemplu:** dacă **n=7**, **k=3** și se citesc cuvintele alăturate, pe ecran se afișează
**paratirisi hiritisi** **(10p.)**

```
isihast
paratirisi
isi
meremetisire
acolisitor
hiritisi
paraponisit
```

**3.** Numim **10-secvență** într-un șir de numere naturale, o succesiune de termeni aflați pe poziții consecutive în șir, cu proprietatea că sunt multipli ai numărului **10**. **Lungimea** secvenței este egală cu numărul de termeni ai săi.
Fișierul **bac.txt** conține un șir de cel mult $10^6$ numere naturale din intervalul $[0,10^9]$, separate prin câte un spațiu. Cel puțin un termen din șir este multiplu al lui **10**. Se cere să se afișeze pe ecran două valori, separate printr-un spațiu, reprezentând lungimea maximă a unei **10**-secvențe din șirul aflat în fișier, respectiv numărul de 10-secvențe cu o astfel de lungime. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.

**Exemplu:** dacă fișierul are conținutul alăturat, se afișează **3 2**

```
7 3 200 100 10 9 6 100 1000 40 1002 20 30
```

**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 59. Informatica — Subiectul III, exercițiul 1

**Sursă: 2020 · Teste Antrenament Oficiale · Test19**

*2020_Informatica_Mate-Info_Teste_Antrenament_Oficiale_C-Cpp_Subiect_Test19.pdf*

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

**1.** Subprogramul **paritate** are doi parametri:

- **n**, prin care primește un număr natural (**n∈[1,10⁹]**);
- **nr** prin care furnizează numărul de divizori naturali ai lui **n** cu aceeași paritate ca **n**.

Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **n=20**, după apel **nr=4** (divizorii lui **20** sunt **1**, **2**, **4**, **5**, **10**, **20**). **(10p.)**

**2.** Într-un tablou bidimensional, cu elemente având valori numai în mulțimea **{0,1}**, numim linii „complementare" două linii cu proprietatea că oricare două elemente ale acestora, aflate pe aceeași coloană, sunt diferite.
Scrieți un program C/C++ care citește de la tastatură trei numere naturale **m**, **n** și **k** (**m,n∈[2,20]**, **k∈[1,m]**) și elementele unui tablou bidimensional cu **m** linii și **n** coloane, numere naturale din mulțimea **{0,1}**. Programul afișează pe ecran mesajul **DA** dacă există cel puțin o linie „complementară" cu linia a **k-a** a acestuia, sau mesajul **NU** în caz contrar.
**Exemplu:** pentru **m=7**, **n=3**, **k=2** și tabloul alăturat se afișează pe ecran mesajul **DA**. **(10p.)**

| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 1 | 0 |
| 0 | 0 | 0 |
| 0 | 0 | 1 |
| 0 | 0 | 1 |
| 1 | 1 | 1 |
| 0 | 0 | 1 |

**3.** Fișierul **bac.txt** conține, **în ordine crescătoare**, cel mult **10⁶** numere naturale din intervalul **[0,10⁹]**, separate prin câte un spațiu. Se cere să se afișeze pe ecran, în ordine strict crescătoare, separate prin câte un spațiu, numerele pare distincte care apar în fișier. Dacă nu există niciun astfel de număr, se afișează pe ecran mesajul **nu exista**. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.
**Exemplu:** dacă fișierul conține numerele **5 5 6 7 8 8 8 8 10 15 16 16 25 25 49**
pe ecran se afișează, în această ordine, numerele **6 8 10 16**
**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 60. Informatica — Subiectul III, exercițiul 1

**Sursă: 2020 · Teste Antrenament Oficiale · Test20**

*2020_Informatica_Mate-Info_Teste_Antrenament_Oficiale_C-Cpp_Subiect_Test20.pdf*

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

**1.** Subprogramul **transformareBaza10** are doi parametri, **b** și **n**, prin care primește câte un număr natural (**b∈[2,10]**, **n∈[0,10⁹]**). Subprogramul returnează suma tuturor produselor de forma $c \cdot b^k$, unde **c** este cifra de pe poziția **k** în scrierea numărului **n**; pozițiile sunt numerotate de la dreapta la stânga, iar cifra unităților este pe poziția **0**.
Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **b=2** și **n=10010**, subprogramul returnează **18** ($18=1 \cdot 2^4+0 \cdot 2^3+0 \cdot 2^2+1 \cdot 2^1+0 \cdot 2^0$). **(10p.)**

**2.** Un text cu cel mult **100** de caractere conține cuvinte și numere, separate prin câte un spațiu. Cuvintele sunt formate numai din litere mici ale alfabetului englez, iar numerele sunt reale, pozitive, cu partea întreagă și partea zecimală separate prin simbolul virgulă, sau numai cu partea întreagă, ca în exemplu. Scrieți un program C/C++ care citește de la tastatură un text de tipul precizat și îl transformă în memorie, înlocuind fiecare număr real cu partea întreagă a acestuia.
**Exemplu:** pentru textul
**partea intreaga a lui 5,75 este egala cu a lui 5,25 si cu a lui 5 si este 5**
se afișează pe ecran
**partea intreaga a lui 5 este egala cu a lui 5 si cu a lui 5 si este 5** **(10p.)**

**3.** Fișierul **bac.txt** conține un șir de cel mult $10^6$ numere naturale din intervalul **[0,10³]**, separate prin câte un spațiu. Se cere să se afișeze pe ecran suma maximă obținută adunând numere cu aceeași paritate, aflate pe poziții consecutive în șirul aflat în fișier. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.
**Exemplu:** dacă fișierul **bac.txt** conține valorile **10 115 1 5 2 2 2 4 7 3 100 20 2 7**
se afișează pe ecran numărul **122**

**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 61. Informatica — Subiectul III, exercițiul 1

**Sursă: 2020 · Toamna August · V05**

*2020_Informatica_Mate-Info_Sesiune_Toamna_August-Septembrie_C-Cpp_Subiect_V05.pdf*

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

**1.** Un număr natural nenul se numește **p-număr** dacă are aceeași paritate cu suma divizorilor săi pozitivi.
**Exemplu:** **10** și **25** sunt p-numere (**10** are aceeași paritate cu **18=1+2+5+10**, iar **25** are aceeași paritate cu **31=1+5+25**).
Subprogramul **kpn**, are trei parametri, **a**, **b** și **k**, prin care primește câte un număr natural din intervalul **[1,10⁶]** (**a≤b**). Subprogramul returnează cel de al **k**-lea p-număr din intervalul **[a,b]** sau **-1**, dacă nu există cel puțin **k** astfel de numere în acest interval. Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **a=27**, **b=50** și **k=3**, atunci subprogramul returnează **34**. **(10p.)**

**2.** Două cuvinte distincte se numesc **în oglindă** dacă fiecare dintre ele se obține prin citirea literelor celuilalt de la dreapta la stânga.
**Exemplu:** **animate** și **etamina** sunt în oglindă, iar pentru cuvântul **reper** nu există un cuvânt cu care să fie în oglindă.
Se consideră un text cu cel mult **100** de caractere, în care cuvintele sunt formate din litere mici ale alfabetului englez și sunt separate prin câte un spațiu. Scrieți un program C/C++ care citește de la tastatură un text de tipul menționat mai sus și îl transformă în memorie, înlocuind fiecare cuvânt cu număr impar de litere cu acel cuvânt cu care el este în oglindă, dacă acesta există, ca în exemplu. Programul afișează pe ecran textul obținut sau mesajul **nu exista**, dacă în text nu s-a înlocuit niciun cuvânt.
**Exemplu**: pentru textul **era o selectie reper de desene animate prezenta**
se obține textul **are o selectie reper de desene etamina prezenta**
iar pentru textul **un reper pentru desene**
se afișează pe ecran mesajul **nu exista** **(10p.)**

**3.** Fișierul **bac.txt** conține un șir de cel mult **10⁶** numere naturale din intervalul **[0,10⁹]**, separate prin câte un spațiu.
Se cere să se afișeze pe ecran, separate printr-un spațiu, două numere naturale **a** și **b** (**a<b**), astfel încât oricare termen al șirului care are exact două cifre să aparțină intervalului **(a,b)**, iar valoarea expresiei **b-a** să fie minimă. Dacă șirul nu are niciun termen de două cifre, pe ecran se afișează mesajul **nu exista**. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al spațiului de memorie utilizat.
**Exemplu:** dacă fișierul conține valorile **7 2 40 5 11 15 10 122 18 350**
se afișează pe ecran numerele **9 41**.
**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 62. Informatica — Subiectul III, exercițiul 1

**Sursă: 2020 · Vara Iunie · V06**

*2020_Informatica_Mate-Info_Sesiune_Vara_Iunie-Iulie_C-Cpp_Subiect_V06.pdf*

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

**1.** Subprogramul **suma** are doi parametri, **a** și **b**, prin care primește câte un număr natural din intervalul **[1,10⁴]**. Subprogramul returnează suma divizorilor naturali comuni lui **a** și **b**.
Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **a=20** și **b=12**, atunci subprogramul returnează valoarea **7** (**1+2+4=7**). **(10p.)**

**2.** Numim **rotire spre stânga** a unui cuvânt format din cel puțin trei litere operația prin care prima sa literă se mută la final, iar toate celelalte litere se mută cu o poziție spre stânga.
**Exemplu:** în urma rotirii spre stânga a cuvântului **ilumina** se obține cuvântul **luminai**.
Un text are cel mult **100** de caractere, iar cuvintele sale sunt formate din litere mici ale alfabetului englez și sunt separate prin câte un spațiu. Scrieți un program C/C++ care citește de la tastatură un text de tipul menționat mai sus și îl transformă în memorie prin rotirea spre stânga a fiecărui cuvânt al său format din cel puțin trei litere, ca în exemplu. Programul afișează pe ecran textul obținut sau mesajul **nu exista**, dacă în text nu există niciun cuvânt de cel puțin trei litere.
**Exemplu**: pentru textul **un palc mic de scolarite ilumina sala**
se afișează pe ecran **un alcp icm de colarites luminai alas** **(10p.)**

**3.** Un șir finit se numește **palindromic** dacă parcurgându-l termen cu termen, de la stânga la dreapta sau de la dreapta la stânga se obține același șir de valori.
**Exemplu:** șirul **12**, **13**, **16**, **13**, **12** este palindromic.
Fișierul **bac.in** conține un șir de cel mult **10⁶** numere naturale din intervalul **[1,10³]**, separate prin câte un spațiu. Se cere să se afișeze pe ecran mesajul **DA**, dacă numerele din șir pot fi rearanjate, astfel încât să formeze un șir palindromic, sau mesajul **NU** în caz contrar. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.
**Exemplu:** dacă fișierul conține numerele **100 30 100 30 500 30 30**
se afișează pe ecran **DA**
**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 63. Informatica — Subiectul III, exercițiul 1

**Sursă: 2021 · Model Oficial**

*2021_Informatica_Mate-Info_Model_Oficial_C-Cpp_Subiect.pdf*

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

**1.** Subprogramul **prime** are trei parametri:
- **n**, prin care primește un număr natural ($n \in [4, 10^9]$);
- **x** și **y**, prin care furnizează cele mai mari două numere prime din intervalul $[1, n)$, $x<y$.

Scrieți definiția completă a subprogramului.
**Exemplu**: dacă **n=49**, în urma apelului **x=43**, **y=47**. **(10p.)**

**2.** Scrieți un program C/C++ care citește de la tastatură două numere naturale din intervalul $[2, 10^2]$, **m** și **n**, și construiește în memorie un tablou bidimensional cu **m** linii și **n** coloane, cu proprietatea că parcurgându-l linie cu linie de sus în jos și fiecare linie de la stânga la dreapta, se obține șirul primelor **m*n pătrate perfecte pare**, ordonat strict descrescător, ca în exemplu.

Elementele tabloului obținut se afișează pe ecran, fiecare linie a tabloului pe câte o linie a ecranului, valorile de pe aceeași linie fiind separate prin câte un spațiu.
**Exemplu:** pentru **m=2**, **n=3** se obține tabloul alăturat. **(10p.)**

```
100  64  36
 16   4   0
```

**3.** Fișierul **cheltuieli.in** are cel mult $10^6$ linii, fiecare linie conținând câte trei numere naturale din intervalul $[1, 10^2]$, reprezentând, în această ordine, date despre câte o achiziție: tipul produsului cumpărat, numărul de produse de acest tip cumpărate, respectiv prețul unui astfel de produs la acel moment. Numerele aflate pe aceeași linie sunt separate prin câte un spațiu.

Se cere să se afișeze pe ecran cea mai mare sumă cheltuită pentru toate produsele de același tip, precum și numărul de tipuri de produse pentru care s-a obținut această sumă. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.
**Exemplu**: dacă fișierul **cheltuieli.in** are conținutul alăturat, se afișează pe ecran: **26 2** (s-a cheltuit suma maximă **26** pentru produsele de tipul **1** și **4**: $26=16 \cdot 1+5 \cdot 2=1 \cdot 10+2 \cdot 8$)

```
4  1 10
1 16  1
4  2  8
2  1  5
1  5  2
```

**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 64. Informatica — Subiectul III, exercițiul 1

**Sursă: 2021 · Simulare Nationala**

*2021_Informatica_Mate-Info_Simulare_Nationala_C-Cpp_Subiect.pdf*

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

**1.** Subprogramul **putere** are un parametru, **n**, prin care primește un număr natural ($n \in [2, 10^9]$). Subprogramul returnează numărul prim care apare la puterea cea mai mică în descompunerea în factori primi a lui **n**. Dacă sunt mai multe astfel de numere, se returnează cel mai mic dintre acestea.
Scrieți definiția completă a subprogramului.
**Exemplu**: dacă **n=880**, subprogramul returnează numărul **5** ($880 = 2^4 \cdot 5 \cdot 11$). **(10p.)**

**2.** Scrieți un program C/C++ care citește de la tastatură numere naturale, în această ordine: **n** ($n \in [2, 20]$), apoi $n \cdot n$ numere din intervalul $[0, 10^2]$, elemente ale unui tablou bidimensional cu **n** linii și **n** coloane, iar la final valoarea **k** ($k \in [2, n]$); programul transformă tabloul în memorie, prin deplasarea circulară spre dreapta, cu câte o poziție, a tuturor valorilor aflate pe linia a **k**-a, în stânga diagonalei principale, ca în exemplu. Elementele tabloului obținut sunt afișate pe ecran, linie cu linie, fiecare linie pe câte o linie a ecranului, cu elementele fiecărei linii separate prin câte un spațiu.

**Exemplu:** pentru **n=6**, **k=5** și tabloul

| | | | | | |
|---|---|---|---|---|---|
| 0 | 1 | 4 | 5 | 3 | 2 |
| 0 | 6 | 0 | 7 | 5 | 0 |
| 4 | 0 | 0 | 6 | 0 | 6 |
| 2 | 0 | 0 | 5 | 0 | 0 |
| 3 | 0 | 6 | 4 | 0 | 8 |
| 0 | 6 | 9 | 0 | 0 | 0 |

se obține tabloul

| | | | | | |
|---|---|---|---|---|---|
| 0 | 1 | 4 | 5 | 3 | 2 |
| 0 | 6 | 0 | 7 | 5 | 0 |
| 4 | 0 | 0 | 6 | 0 | 6 |
| 2 | 0 | 0 | 5 | 0 | 0 |
| 4 | 3 | 0 | 6 | 0 | 8 |
| 0 | 6 | 9 | 0 | 0 | 0 |

**(10p.)**

**3.** La proiectarea unui site web se utilizează elemente grafice realizate pe baza unor modele. Fiecare model este de formă pătrată și oricare două modele distincte au dimensiuni diferite ale laturilor. Toate elementele grafice realizate pe baza unui anumit model au aceeași formă și aceleași dimensiuni ca ale acestuia. În vederea asigurării elementelor grafice necesare, pentru fiecare model dintre cele utilizate se plătește o taxă unică de proiectare, de **10** lei, iar pentru fiecare element grafic realizat pe baza acelui model se plătește o sumă în lei, egală cu valoarea suprafeței acestuia (aria pătratului), calculată în centimetri pătrați. Fișierul **bac.in** conține un șir de cel mult $10^6$ numere naturale din intervalul **[1,10]**, separate prin câte un spațiu, reprezentând dimensiunile laturilor tuturor elementelor grafice utilizate, date în centimetri; fiecare termen al șirului corespunde unui element grafic distinct. Se cere să se afișeze pe ecran suma totală plătită pentru asigurarea elementelor grafice necesare. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.
**Exemplu:** dacă fișierul conține numerele **1 7 2 1 1 2 1 7 2**
se afișează pe ecran valoarea **144**
(**10** lei pentru modelul de lățime **1** cm și câte $1 \cdot 1$ lei pentru fiecare dintre cele patru elemente grafice care îl au la bază, **10** lei pentru modelul de lățime **2** cm și câte $2 \cdot 2$ lei pentru fiecare dintre cele trei elemente grafice care îl au la bază, respectiv **10** lei pentru modelul de lățime **7** cm și câte $7 \cdot 7$ lei pentru fiecare dintre cele două elemente grafice care îl au la bază).
**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 65. Informatica — Subiectul III, exercițiul 1

**Sursă: 2021 · Speciala C · V07**

*2021_Informatica_Mate-Info_Sesiune_Speciala_C-Cpp_Subiect_V07.pdf*

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

**1.** Două numere se numesc **oglindite** dacă fiecare se obține din celălalt, prin parcurgerea cifrelor acestuia de la dreapta la stânga. Două numere se numesc **impar-oglindite** dacă numerele obținute din acestea, prin îndepărtarea tuturor cifrelor lor pare, sunt oglindite.
Subprogramul **imog** are trei parametri:
- **x** și **y**, prin care primește câte un număr natural din intervalul $[0,10^9]$;
- **rez**, prin care furnizează valoarea 1 dacă **x** și **y** sunt impar-oglindite sau valoarea 0 în caz contrar.

Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **x=523** și **y=84356**, după apel **rez=1**,
iar dacă **x=523** și **y=84536** sau **x=523** și **y=84576** sau **x=40** și **y=86**, după apel **rez=0**. **(10p.)**

**2.** Scrieți un program C/C++ care citește de la tastatură numere naturale: **n** ($n\in[2,20]$), **k** ($k\in[2,n]$) și $n\cdot n$ numere din intervalul $[0,10^9]$, elemente ale unui tablou bidimensional cu **n** linii și **n** coloane. Programul transformă tabloul în memorie, interschimbând șirul elementelor situate pe linia a **k**-a, în stânga diagonalei principale, parcurse de la stânga la dreapta, cu șirul elementelor situate pe coloana a **k**-a, deasupra diagonalei principale, parcurse de sus în jos, ca în exemplu. Elementele tabloului obținut sunt afișate pe ecran, fiecare linie pe câte o linie a ecranului, cu elementele fiecărei linii separate prin câte un spațiu.

**Exemplu:** pentru **n=5**, **k=4** și tabloul

| | | | | |
|---|---|---|---|---|
| 2 | 4 | 3 | 5 | 6 |
| 8 | 0 | 9 | 8 | 7 |
| 2 | 6 | 9 | 0 | 5 |
| 6 | 1 | 3 | 6 | 9 |
| 7 | 3 | 9 | 4 | 2 |

se obține tabloul

| | | | | |
|---|---|---|---|---|
| 2 | 4 | 3 | 6 | 6 |
| 8 | 0 | 9 | 1 | 7 |
| 2 | 6 | 9 | 3 | 5 |
| 5 | 8 | 0 | 6 | 9 |
| 7 | 3 | 9 | 4 | 2 |

**(10p.)**

**3.** Se consideră șirul **1**, **3**, **7**, **13**, **21**, **31**, **43** . . . definit astfel: $f_0=1$, iar $f_n=f_{n-1}+2\cdot n$, dacă $n\geq 1$ (unde **n** este un număr natural).
Se citesc de la tastatură două numere naturale din intervalul $[1,10^9]$, **x** și **y** (**x<y**), reprezentând doi termeni aflați pe poziții consecutive în șirul dat, și se cere să se scrie în fișierul text **bac.out**, separați prin câte un spațiu, toți termenii șirului mai mici sau egali cu **y**, în ordine inversă a apariției lor în șir. Proiectați un algoritm eficient din punctul de vedere al spațiului de memorie și al timpului de executare.
**Exemplu:** dacă **x=21** și **y=31**, fișierul conține valorile
**31 21 13 7 3 1**

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

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

## 66. Informatica — Subiectul III, exercițiul 1

**Sursă: 2021 · Teste Antrenament Oficiale · Test01**

*2021_Informatica_Mate-Info_Teste_Antrenament_Oficiale_C-Cpp_Subiect_Test01.pdf*

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

**1.** Subprogramul **divX** are doi parametri, **n** și **x**, prin care primește câte un număr natural din intervalul `[2,50]`. Subprogramul afișează pe ecran, în ordine descrescătoare, separate prin câte un spațiu, primele **n** numere naturale nenule divizibile cu **x**.
Scrieți definiția completă a subprogramului.
**Exemplu**: dacă **n=4** și **x=15** în urma apelului se afișează numerele `60 45 30 15` **(10p.)**

**2.** Scrieți un program C/C++ care citește de la tastatură numărul natural **n** ($n \in [2,10^2]$) și elementele unui tablou bidimensional cu **n** linii și **n** coloane, numere naturale din intervalul $[0,10^9]$.
Programul afișează pe ecran, separate prin câte un spațiu, elementele primului pătrat concentric, parcurs în sens invers al acelor de ceasornic, începând din colțul său stânga-sus, ca în exemplu. Primul pătrat concentric este format din prima și ultima linie, prima și ultima coloană a tabloului.
**Exemplu:** pentru **n=5** și tabloul alăturat, se afișează pe ecran numerele
`1 2 3 4 5 6 7 8 9 0 2 4 6 8 1 3` **(10p.)**

| | | | | |
|---|---|---|---|---|
| 1 | 3 | 1 | 8 | 6 |
| 2 | 9 | 2 | 7 | 4 |
| 3 | 5 | 8 | 5 | 2 |
| 4 | 1 | 6 | 3 | 0 |
| 5 | 6 | 7 | 8 | 9 |

**3.** Fișierul **bac.in** conține cel mult $10^6$ numere naturale din intervalul $[0,10^9]$, separate prin câte un spațiu. Se cere să se afișeze pe ecran, în ordine descrescătoare, cele mai mari două numere de două cifre distincte care **NU** se află în fișier. Numerele afișate sunt separate printr-un spațiu, iar dacă nu există două astfel de numere, se afișează pe ecran mesajul **nu exista**. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.
**Exemplu:** dacă fișierul **bac.in** conține numerele `12 235 123 67 98 6 96 94 123 67 98 100`
se afișează pe ecran, în această ordine, numerele **97 95**.
**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 67. Informatica — Subiectul III, exercițiul 1

**Sursă: 2021 · Teste Antrenament Oficiale · Test02**

*2021_Informatica_Mate-Info_Teste_Antrenament_Oficiale_C-Cpp_Subiect_Test02.pdf*

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

**1.** Subprogramul **factori** are doi parametri, **n** și **m**, prin care primește câte un număr natural din intervalul $[1,10^9]$. Subprogramul returnează numărul valorilor prime care apar la aceeași putere atât în descompunerea în factori primi a lui **n**, cât și în descompunerea în factori primi a lui **m**.
Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **n=16500** și **m=10780**, atunci subprogramul returnează **2** ($16500=\underline{2^2}\cdot 3\cdot 5^3\cdot \underline{11}$, $10780=\underline{2^2}\cdot 5\cdot 7^2\cdot \underline{11}$). **(10p.)**

**2.** Scrieți un program C/C++ care citește de la tastatură un număr natural, **n** ($n\in[2,20]$), și construiește în memorie un tablou bidimensional cu **n** linii și **n** coloane, astfel încât fiecare linie a sa are proprietățile:
- elementul situat pe diagonala secundară are valoarea **n**;
- începând de la diagonala secundară, spre stânga, elementele formează un șir strict descrescător de numere consecutive, iar începând de la diagonala secundară, spre dreapta, elementele formează tot un șir strict descrescător de numere consecutive.

Programul afișează pe ecran tabloul construit, fiecare linie a tabloului pe câte o linie a ecranului, cu elementele aflate pe aceeași linie separate prin câte un spațiu.
**Exemplu:** dacă **n=5** se afișează pe ecran tabloul alăturat. **(10p.)**

```
1 2 3 4 5
2 3 4 5 4
3 4 5 4 3
4 5 4 3 2
5 4 3 2 1
```

**3.** Fișierul **bac.in** conține un șir de cel mult $10^6$ numere întregi din intervalul $[-10^9,10^9]$, separate prin câte un spațiu. Cel puțin un număr din șir este pozitiv.
Se cere să se afișeze pe ecran lungimea maximă a unei secvențe a șirului care fie începe, fie se încheie cu un număr pozitiv. O secvență este formată din termeni aflați pe poziții consecutive în șir, iar lungimea secvenței este egală cu numărul de termeni ai acesteia. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.
**Exemplu:** dacă fișierul conține numerele **-15 -7 4 -7 21 -5 -200 -26 52 -24 -7 -9 -20** pe ecran se afișează **11** (corespunzător secvenței **4 -7 21 -5 -200 -26 52 -24 -7 -9 -20**).
**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 68. Informatica — Subiectul III, exercițiul 1

**Sursă: 2021 · Teste Antrenament Oficiale · Test03**

*2021_Informatica_Mate-Info_Teste_Antrenament_Oficiale_C-Cpp_Subiect_Test03.pdf*

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

**1.** Subprogramul **suma** are un singur parametru, **n**, prin care primește un număr natural ($n \in [1, 10^6]$). Subprogramul returnează suma divizorilor pozitivi ai lui **n** care nu sunt primi.
Scrieți definiția completă a subprogramului.
**Exemplu:** pentru **n=12** subprogramul returnează **23** (**23=1+4+6+12**). **(10p.)**

**2.** O sală de spectacole are mai multe rânduri de scaune, toate rândurile având același număr de locuri. O persoană poate rezerva mai multe locuri, dar cel mult câte unul pe un rând. Configurația sălii este memorată ca un tablou bidimensional, în care prima **coloană** corespunde primului rând de scaune, iar ultima **coloană** corespunde ultimului rând de scaune. Locurile libere sunt notate cu **0**, iar cele rezervate sunt notate cu numere naturale nenule, astfel încât toate locurile rezervate de o persoană sunt notate cu același număr, iar cele rezervate de persoane distincte sunt notate cu numere distincte.
Scrieți un program C/C++ care citește de la tastatură numerele naturale din intervalul **[2,20]**, **m** și **n**, apoi **m·n** numere naturale din intervalul **[0,400]**, elemente ale tabloului bidimensional cu **m** linii și **n** coloane, reprezentând configurația sălii. Programul verifică dacă există persoane care au rezervat locuri atât pe primul, cât și pe ultimul rând al sălii, și afișează pe ecran numerele cu care sunt notate aceste locuri, ca în exemplu.
Numerele sunt afișate într-o ordine oarecare, separate prin câte un spațiu, iar dacă nu există astfel de locuri se afișează pe ecran mesajul **nu exista**.
**Exemplu:** pentru **m=5**, **n=4** și tabloul alăturat se afișează pe ecran, nu neapărat în această ordine, numerele **6 8** **(10p.)**

| | | | |
|---|---|---|---|
| 3 | 7 | 0 | 0 |
| 0 | 0 | 5 | 8 |
| 6 | 2 | 0 | 1 |
| 0 | 3 | 0 | 6 |
| 8 | 0 | 6 | 4 |

**3.** Două numere naturale sunt numite **z-prietene** dacă au aceeași cifră a zecilor.
Fișierul **bac.in** conține un șir de cel mult $10^6$ numere naturale din intervalul $[10, 10^9]$, separate prin câte un spațiu. Se cere să se afișeze pe ecran pozițiile din șir pe care se află termeni precedați de un număr maxim de valori z-prietene cu ei. Numerele afișate sunt separate prin câte un spațiu. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.
**Exemplu:** dacă fișierul conține numerele **726 358 98 157 20 49 128 879 659 271**
pe ecran se afișează numerele **7 9** (termenii **128**, respectiv **659** respectă proprietatea cerută).
**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 69. Informatica — Subiectul III, exercițiul 1

**Sursă: 2021 · Teste Antrenament Oficiale · Test04**

*2021_Informatica_Mate-Info_Teste_Antrenament_Oficiale_C-Cpp_Subiect_Test04.pdf*

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

**1.** Un joc online cu **n** jetoane poate fi jucat de un grup de **k** (**k≥2**) jucători, numai dacă toate cele **n** jetoane pot fi distribuite în mod egal celor **k** jucători.
Subprogramul **joc** are un singur parametru, **n**, prin care primește un număr natural (**n∈[2,10⁴]**), reprezentând numărul de jetoane ale unui joc de tipul precizat. Subprogramul returnează numărul valorilor distincte pe care le poate avea **k** pentru acest joc.
Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **n=12**, atunci subprogramul returnează numărul **5** (cele **12** jetoane se pot distribui în mod egal pentru o grupă de **2** jucători, de **3** jucători, de **4** jucători, de **6** jucători sau de **12** jucători).
**(10p.)**

**2.** Scrieţi un program C/C++ care citeşte de la tastatură numere naturale din intervalul **[2,10²]**: **m**, **n** şi elementele unui tablou bidimensional cu **m** linii şi **n** coloane. Programul afișează pe ecran, pentru fiecare linie a sa, cea mai mare dintre valorile strict mai mici decât **21** memorate în aceasta, sau mesajul **nu exista**, dacă nu există nicio astfel de valoare pe linia respectivă. Numerele, respectiv mesajele, sunt afișate pe linii separate ale ecranului, în ordinea liniilor corespunzătoare din tablou.

**Exemplu:** pentru **m=3**, **n=5** și tabloul alăturat se afişează pe ecran **(10p.)**

```
16
nu exista
9
```

| | | | | |
|---|---|---|---|---|
| 6 | 16 | 21 | 4 | 90 |
| 92 | 26 | 36 | 95 | 80 |
| 5 | 2 | 9 | 7 | 3 |

**3.** Fişierele **bac1.txt** și **bac2.txt** conţin numere naturale din intervalul **[1,10⁵]**: fișierul **bac1.txt** conține pe prima linie un număr **n1**, iar pe a doua linie un șir de **n1** numere, iar fișierul **bac2.txt** conține pe prima linie un număr **n2**, iar pe a doua linie un șir de **n2** numere. Numerele aflate pe aceeași linie a unui fișier sunt ordonate crescător și sunt separate prin câte un spațiu.
Se cere să se afișeze pe ecran, în ordine crescătoare, separate prin câte un spațiu, numerele divizibile cu **5** care se găsesc doar în unul dintre șirurile aflate în cele două fișiere. Dacă nu există niciun astfel de număr, se afișează pe ecran mesajul **nu exista**. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate şi al timpului de executare.

**Exemplu**: dacă fişierul **bac1.txt** conţine numerele

```
7
1 2 3 4 7 20 60
```

iar fişierul **bac2.txt** conţine numerele

```
9
3 5 7 8 9 10 12 20 24
```

pe ecran se afișează, în această ordine, numerele **5 10 60**.
**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 70. Informatica — Subiectul III, exercițiul 1

**Sursă: 2021 · Teste Antrenament Oficiale · Test05**

*2021_Informatica_Mate-Info_Teste_Antrenament_Oficiale_C-Cpp_Subiect_Test05.pdf*

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

**1.** Subprogramul **identice** are un singur parametru, **n**, prin care primește un număr natural (**n∈[10,10⁹]**). Subprogramul returnează valoarea **1**, dacă numărul **n** are toate cifrele egale, sau valoarea **0** în caz contrar. Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **n=2222**, subprogramul returnează valoarea **1**, iar dacă **n=212**, subprogramul returnează valoarea **0**. **(10p.)**

**2.** Scrieţi un program C/C++ care citeşte de la tastatură două numere naturale din intervalul **[2,10²]**, **m** și **n**, şi construieşte în memorie un tablou bidimensional cu **m** linii şi **n** coloane, numerotate începând cu **1**, astfel încât elementul de pe linia **i** şi coloana **j** primeşte ca valoare ultima cifră a produsului **i·j**.
Programul afişează pe ecran elementele tabloului obținut, linie cu linie, fiecare linie a tabloului pe câte o linie a ecranului, cu valorile aflate pe aceeaşi linie separate prin câte un spaţiu.
**Exemplu:** dacă **m=4** și **n=5** se afişează pe ecran tabloul alăturat. **(10p.)**

```
1 2 3 4 5
2 4 6 8 0
3 6 9 2 5
4 8 2 6 0
```

**3.** Fişierul **bac.txt** conține numere naturale din intervalul **[2,10⁶]**: pe prima linie **n**, iar pe a doua linie un șir de **n** numere, separate prin câte un spațiu.
Se cere să se afișeze pe ecran, pentru fiecare număr natural **i** (**i∈[1,n]**), cea mai mare dintre primele **i** valori ale șirului aflat în fișier. Numerele afișate sunt separate prin câte un spațiu. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate şi al timpului de executare.

**Exemplu:** dacă fișierul are conținutul alăturat, se afișează pe ecran

```
4 6 6 7 8 8 8 8 8 9 10 10
```

```
12
4 6 3 7 8 1 6 2 7 9 10 8
```

**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 71. Informatica — Subiectul III, exercițiul 1

**Sursă: 2021 · Teste Antrenament Oficiale · Test06**

*2021_Informatica_Mate-Info_Teste_Antrenament_Oficiale_C-Cpp_Subiect_Test06.pdf*

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

**1.** Subprogramul **numar** are trei parametri:

- **n** și **c**, prin care primește câte un număr natural ($n \in [0, 10^9]$, $c \in [0, 9]$);
- **m**, prin care furnizează numărul obținut din **n**, prin eliminarea din acesta a tuturor cifrelor egale cu **c**, sau **-1** dacă toate cifrele lui **n** sunt egale cu **c**. Cifrele nule nesemnificative sunt ignorate, ca în exemplu.

Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **n=50752** sau **n=72** și **c=5**, după apel **m=72**, dacă **n=500** și **c=5**, după apel **m=0**, iar dacă **n=55** și **c=5**, după apel **m=-1**. **(10p.)**

**2.** Scrieți un program C/C++ care citește de la tastatură un număr natural, **n** ($n \in [3, 20]$), și construiește în memorie un tablou bidimensional cu **n** linii și **n** coloane, având proprietățile:

- toate elementele situate pe diagonala secundară sunt nule;
- prima linie conține un șir strict descrescător de numere consecutive, iar ultima linie conține un șir strict crescător de numere consecutive;
- fiecare dintre celelalte linii conține, începând cu prima poziție, până la diagonala secundară inclusiv, de la stânga la dreapta, un șir strict descrescător de numere consecutive, iar începând de la diagonala secundară, inclusiv, până la ultima poziție, de la stânga la dreapta, un șir strict crescător de numere consecutive.

Programul afișează pe ecran tabloul construit, fiecare linie a tabloului pe câte o linie a ecranului, cu elementele aflate pe aceeași linie separate prin câte un spațiu.
**Exemplu:** dacă **n=4** se afișează pe ecran tabloul alăturat. **(10p.)**

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

**3.** Fișierul **bac.in** conține un șir de cel puțin patru și cel mult $10^5$ numere întregi nenule din intervalul $[-10^9, 10^9]$, dintre care trei sunt negative, iar restul pozitive. Numerele sunt separate prin câte un spațiu. O secvență este formată din termeni aflați pe poziții consecutive în șir, iar lungimea secvenței este egală cu numărul de termeni ai acesteia.
Se cere să se afișeze pe ecran lungimea unei secvențe din șirul aflat în fișier care conține o singură valoare negativă și un număr maxim de valori pozitive. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.
**Exemplu:** dacă fișierul conține numerele **15 21 -61 9 870 -23 11 5 8 -81 5 14**
pe ecran se afișează **6** (corespunzător secvențelor **9 870 -23 11 5 8** sau **11 5 8 -81 5 14**).

**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 72. Informatica — Subiectul III, exercițiul 1

**Sursă: 2021 · Teste Antrenament Oficiale · Test07**

*2021_Informatica_Mate-Info_Teste_Antrenament_Oficiale_C-Cpp_Subiect_Test07.pdf*

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

**1.** Subprogramul **afisare** are trei parametri:

- **x** și **y**, prin care primește câte un număr natural din intervalul $[0,10^6]$ ($x \leq y$);
- **k**, prin care primește un număr natural ($k \in [2,10^2]$).

Subprogramul afișează pe ecran, în ordine strict crescătoare, numerele din intervalul **[x,y]**, în secvențe de câte **k**, cu excepția ultimei secvențe care poate conține mai puțin de **k** numere. Fiecare secvență se încheie cu câte un simbol *, iar numerele și simbolurile sunt separate prin câte un spațiu, ca în exemplu.
Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **x=11**, **y=21** și **k=4** se afișează pe ecran numerele de mai jos, în acest format.
`11 12 13 14 * 15 16 17 18 * 19 20 21 *` **(10p.)**

**2.** Scrieți un program C/C++ care citește de la tastatură un număr natural, **x** ($x \in [1,10^9]$), și construiește în memorie un tablou bidimensional, pentru care atât numărul de linii, cât și numărul de coloane sunt egale cu numărul de cifre ale lui **x**, iar elementele fiecărei linii au ca valori cifrele lui **x**, în ordine, ca în exemplu.

Elementele tabloului obținut sunt afișate pe ecran, linie cu linie, fiecare linie a tabloului pe câte o linie a ecranului, cu elementele de pe aceeași linie separate prin câte un spațiu.
**Exemplu**: dacă **x=1359**, se afișează tabloul alăturat. **(10p.)**

```
1 3 5 9
1 3 5 9
1 3 5 9
1 3 5 9
```

**3.** Fișierul **bac.txt** conține cel mult $10^6$ cifre, separate prin câte un spațiu.
Se cere să se afișeze pe ecran, separate prin câte un spațiu, toate cifrele pare care apar în fișier sau mesajul **nu exista**, dacă nu există astfel de cifre. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.
**Exemplu:** dacă fișierul conține cifrele `3 3 0 8 2 1 2 1 3 7 1 5 2 7 1 0 3 2 3`
pe ecran se afișează, de exemplu în ordine crescătoare, cifrele `0 0 2 2 2 2 8`
**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++corespunzător algoritmului proiectat. **(8p.)**

## 73. Informatica — Subiectul III, exercițiul 1

**Sursă: 2021 · Teste Antrenament Oficiale · Test08**

*2021_Informatica_Mate-Info_Teste_Antrenament_Oficiale_C-Cpp_Subiect_Test08.pdf*

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

**1.** Subprogramul **nrfp** are doi parametri:

- **n**, prin care primește un număr natural ($n \in [2, 10^5]$);
- **m**, prin care furnizează numărul din intervalul închis **[2,n]** care are cei mai mulți factori primi; dacă există mai multe numere cu această proprietate, subprogramul îl returnează pe cel mai mare dintre ele.

Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **n=100** atunci, în urma apelului, **m=90**. **(10p.)**

**2.** Scrieți un program C/C++ care citește de la tastatură numărul natural **n** ($n \in [5,50]$) și elementele unui tablou bidimensional cu **n** linii și **n** coloane, numere naturale din intervalul $[0, 10^2]$. Programul afișează pe ecran suma numerelor din zona delimitată de cele două diagonale și ultima coloană a tabloului, ca în exemplu.
**Exemplu:** pentru **n=7** și tabloul alăturat, se afișează pe ecran **12**.
**(10p.)**

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 5 | 6 |
| 7 | 8 | 9 | 0 | 3 | 1 | 2 |
| 4 | 6 | 8 | 0 | 1 | 1 | 3 |
| 8 | 6 | 3 | 6 | 2 | 4 | 7 |
| 5 | 7 | 9 | 2 | 2 | 5 | 8 |
| 1 | 4 | 7 | 0 | 5 | 3 | 6 |
| 9 | 2 | 5 | 8 | 5 | 9 | 1 |

**3.** Fișierul **bac.txt** conține un șir de cel mult $10^6$ numere naturale din intervalul $[0, 10^9]$.
Se cere să se determine și să se afișeze pe ecran, separate printr-un spațiu, ultimele două numere impare (nu neapărat distincte) din șirul aflat în fișier, sau mesajul **nu exista**, dacă nu există două astfel de numere. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.
**Exemplu:** dacă fișierul conține valorile **122 1635 628 1413 1647 900 3001 4252**
se afișează pe ecran **1647 3001**

**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 74. Informatica — Subiectul III, exercițiul 1

**Sursă: 2021 · Teste Antrenament Oficiale · Test09**

*2021_Informatica_Mate-Info_Teste_Antrenament_Oficiale_C-Cpp_Subiect_Test09.pdf*

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

**1.** Subprogramul **divizor** are patru parametri:
- **a**, **b** și **k**, prin care primește câte un număr natural (**a**$\in$**[0,10⁹]**, **b**$\in$**[a,10⁹]**, **k**$\in$**[1,9]**);
- **nr**, prin care furnizează numărul de valori naturale din intervalul **[a,b]** care sunt divizibile cu **k** și au ultima cifră egală cu **k**. Scrieți definiția completă a subprogramului.

**Exemplu**: dacă **a=3**, **b=50** și **k=4**, în urma apelului, **nr=3** (pentru numerele **4**, **24**, **44**). **(10p.)**

**2.** Într-un text cu cel mult **10²** caractere, cuvintele sunt formate din litere mici ale alfabetului englez și sunt separate prin câte un spațiu. Scrieți un program C/C++ care citește de la tastatură un text de tipul precizat și afișează pe ecran, pe linii separate, toate cuvintele sale care conțin o singură vocală distinctă, ca în exemplu.

Dacă nu există niciun astfel de cuvânt, se afișează pe ecran mesajul **nu exista**. Se consideră vocale literele din mulțimea **a**, **e**, **i**, **o**, **u**.

**Exemplu**: pentru textul **a plantat cinci lalele visinii sau rosii** se afișează pe ecran, nu neapărat în această ordine, cuvintele alăturate. **(10p.)**

```
a
plantat
cinci
visinii
```

**3.** Fișierul **numere.txt** conține cel mult **10⁵** numere naturale din intervalul **[1,10⁹]**, câte unul pe fiecare linie. Se cere să se afișeze pe ecran cel mai mare număr care se poate forma cu toate cifrele care apar în numerele din fișier, ca în exemplu.

Proiectați un algoritm eficient din punctul de vedere al timpului de executare.

**Exemplu**: dacă fișierul are conținutul alăturat, se afișează

**9988887766333220**

```
263
39628
79
887308
```

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

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

## 75. Informatica — Subiectul III, exercițiul 1

**Sursă: 2021 · Teste Antrenament Oficiale · Test10**

*2021_Informatica_Mate-Info_Teste_Antrenament_Oficiale_C-Cpp_Subiect_Test10.pdf*

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

**1.** Numerele naturale **x** și **y** sunt numite **în armonie** dacă suma lor aparține intervalului deschis definit de suma divizorilor lui **x**, respectiv suma divizorilor lui **y**.
Subprogramul **armonie** are doi parametri, **x** şi **y**, prin care primește câte un număr natural din intervalul **[1,10⁶]**. Subprogramul returnează valoarea **1**, dacă **x** și **y** sunt în armonie, sau valoarea **0** în caz contrar.
Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **x=8**, iar **y=12** subprogramul returnează **1** (**1+2+4+8=15**, **1+2+4+6+12=25**, iar **8+12=20∈(15,25)**), iar dacă **x=8** și **y=13**, subprogramul returnează **0** (**1+2+4+8=15**, **1+13=14**, iar **8+13=21∉(14,15)**). **(10p.)**

**2.** O valoare **filtrează** două șiruri dacă există doi termeni care au acea valoare, unul fiind în primul șir, iar celălalt în al doilea șir.
Scrieți un program C/C++ care citește de la tastatură numere naturale din intervalul **[2,20]**: **m**, **n** și elementele unui tablou bidimensional cu **m** linii și **n** coloane, cu proprietatea că nu există două elemente egale situate pe aceeași linie sau pe aceeași coloană.
Programul afișează pe ecran valorile care pot filtra șirul format din primele **n-1** elemente de pe prima linie, respectiv șirul format din ultimele **m-1** elemente ale ultimei coloane a tabloului, ca în exemplu. Valorile sunt afișate într-o ordine oarecare, separate prin câte un spațiu, sau mesajul **nu exista**, dacă nu există astfel de valori.
**Exemplu:** pentru **m=5**, **n=4** și tabloul alăturat, se afișează pe ecran, nu neapărat în această ordine, numerele **4 7** **(10p.)**

| | | | |
|---|---|---|---|
| **4** | 5 | **7** | 2 |
| 2 | 7 | 3 | 6 |
| 7 | 6 | 4 | 0 |
| 6 | 9 | 8 | **7** |
| 8 | 0 | 5 | **4** |

**3.** Fişierul **bac.txt** conține un șir de cel mult **10⁵** numere naturale din intervalul **[1,10⁹]**, separate prin câte un spaţiu.
Se cere să se afişeze pe ecran cea mai mică poziţie pe care ar putea-o ocupa primul termen al şirului aflat în fişier în şirul format cu aceleași valori, ordonat crescător. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate şi al timpului de executare.
**Exemplu:** dacă fişierul conţine numerele **15 7 15 17 6 4**
se afişează pe ecran **4** (**15** se află pe a patra și pe a cincea poziție în şirul **4**, **6**, **7**, **15**, **15**, **17**).
**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 76. Informatica — Subiectul III, exercițiul 1

**Sursă: 2021 · Teste Antrenament Oficiale · Test11**

*2021_Informatica_Mate-Info_Teste_Antrenament_Oficiale_C-Cpp_Subiect_Test11.pdf*

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

**1.** Două numere se numesc **oglindite** dacă fiecare se obține din celălalt, prin parcurgerea cifrelor acestuia de la dreapta la stânga. Două numere se numesc **impar-oglindite** dacă numerele obținute din acestea, prin îndepărtarea tuturor cifrelor lor pare, sunt oglindite.
Subprogramul **imog** are trei parametri:
- **x** și **y**, prin care primește câte un număr natural din intervalul $[0,10^9]$;
- **rez**, prin care furnizează valoarea 1 dacă **x** și **y** sunt impar-oglindite sau valoarea 0 în caz contrar.

Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **x=523** și **y=84356**, după apel **rez=1**,
iar dacă **x=523** și **y=84536** sau **x=523** și **y=84576** sau **x=40** și **y=86**, după apel **rez=0**. **(10p.)**

**2.** Scrieți un program C/C++ care citește de la tastatură numere naturale: **n** ($n\in[2,20]$), **k** ($k\in[2,n]$) și $n\cdot n$ numere din intervalul $[0,10^9]$, elemente ale unui tablou bidimensional cu **n** linii și **n** coloane. Programul transformă tabloul în memorie, interschimbând șirul elementelor situate pe linia a **k**-a, în stânga diagonalei principale, parcurse de la stânga la dreapta, cu șirul elementelor situate pe coloana a **k**-a, deasupra diagonalei principale, parcurse de sus în jos, ca în exemplu. Elementele tabloului obținut sunt afișate pe ecran, fiecare linie pe câte o linie a ecranului, cu elementele fiecărei linii separate prin câte un spațiu.

**Exemplu:** pentru **n=5**, **k=4** și tabloul

| | | | | |
|---|---|---|---|---|
| 2 | 4 | 3 | 5 | 6 |
| 8 | 0 | 9 | 8 | 7 |
| 2 | 6 | 9 | 0 | 5 |
| 6 | 1 | 3 | 6 | 9 |
| 7 | 3 | 9 | 4 | 2 |

se obține tabloul

| | | | | |
|---|---|---|---|---|
| 2 | 4 | 3 | 6 | 6 |
| 8 | 0 | 9 | 1 | 7 |
| 2 | 6 | 9 | 3 | 5 |
| 5 | 8 | 0 | 6 | 9 |
| 7 | 3 | 9 | 4 | 2 |

**(10p.)**

**3.** Se consideră șirul **1**, **3**, **7**, **13**, **21**, **31**, **43** . . . definit astfel: $f_0=1$, iar $f_n=f_{n-1}+2\cdot n$, dacă $n\geq 1$ (unde **n** este un număr natural).
Se citesc de la tastatură două numere naturale din intervalul $[1,10^9]$, **x** și **y** (**x<y**), reprezentând doi termeni aflați pe poziții consecutive în șirul dat, și se cere să se scrie în fișierul text **bac.out**, separați prin câte un spațiu, toți termenii șirului mai mici sau egali cu **y**, în ordine inversă a apariției lor în șir. Proiectați un algoritm eficient din punctul de vedere al spațiului de memorie și al timpului de executare.
**Exemplu:** dacă **x=21** și **y=31**, fișierul conține valorile
**31 21 13 7 3 1**

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

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

## 77. Informatica — Subiectul III, exercițiul 1

**Sursă: 2021 · Teste Antrenament Oficiale · Test12**

*2021_Informatica_Mate-Info_Teste_Antrenament_Oficiale_C-Cpp_Subiect_Test12.pdf*

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

**1.** Un număr `y` este numit **frate mai mare** al unui număr `x` dacă `x` și `y` au același număr de cifre și fiecare cifră a lui `y` se poate obține din cifra aflată pe aceeași poziție în `x` adunând la aceasta valoarea `1`.
Subprogramul `frate` are doi parametri:

- `x`, prin care primește un număr natural (`x`$\in$`[0,10⁹]`);
- `y`, prin care furnizează fratele mai mare al lui `x`, sau `-1`, dacă nu se poate obține un astfel de număr.

Scrieți definiția completă a subprogramului.
**Exemplu:** dacă `x=1027`, după apel `y=2138`, iar dacă `x=9027`, după apel `y=-1`. **(10p.)**

**2.** Scrieți un program C/C++ care citește de la tastatură numere naturale: `n` (`n`$\in$`[2,20]`), `k` (`k`$\in$`[1,n]`) și `n·n` numere din intervalul `[0,10⁹]`, elemente ale unui tablou bidimensional cu `n` linii și `n` coloane. Programul transformă tabloul în memorie, deplasând circular spre dreapta, cu câte o poziție, toate elementele situate pe linia a `k`-a, în stânga diagonalei secundare, ca în exemplu. Elementele tabloului obținut sunt afișate pe ecran, fiecare linie pe câte o linie a ecranului, cu elementele fiecărei linii separate prin câte un spațiu.

**Exemplu:** pentru `n=5`, `k=2` și tabloul

```
2 3 4 5 6
2 4 6 8 0
7 8 9 0 1
3 5 7 9 1
7 3 8 5 6
```

se obține tabloul

```
2 3 4 5 6
6 2 4 8 0
7 8 9 0 1
3 5 7 9 1
7 3 8 5 6
```

**(10p.)**

**3.** Fișierul `bac.txt` conține un șir de cel mult $10^5$ numere naturale din intervalul `[1,10⁹]`, separate prin câte un spațiu. Se cere să se afișeze pe ecran cea mai mare poziție pe care ar putea-o ocupa primul termen al șirului aflat în fișier în șirul format cu aceleași valori, ordonat descrescător. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.
**Exemplu:** dacă fișierul `bac.txt` conține numerele `15 7 15 17 6 4 21`
se afișează pe ecran `4` (`15` se află pe a treia și pe a patra poziție în șirul `21`, `17`, `15`, `15`, `7`, `6`, `4`).
**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 78. Informatica — Subiectul III, exercițiul 1

**Sursă: 2021 · Toamna August · V04**

*2021_Informatica_Mate-Info_Sesiune_Toamna_August-Septembrie_C-Cpp_Subiect_V04.pdf*

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

**1.** Un număr natural **n** se numește **cub perfect** dacă există un număr natural **b**, astfel încât $n=b^3$.
Subprogramul **cuburi** are un singur parametru, **n**, prin care primește un număr natural ($n \in [1,10^3]$). Subprogramul afișează pe ecran, separate prin câte un spațiu, în ordine descrescătoare, primele **n** cuburi perfecte nenule.
Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **n=5** atunci, după apel, se afișează pe ecran numerele
**125 64 27 8 1** **(10p.)**

**2.** Scrieți un program C/C++ care citește de la tastatură două numere naturale din intervalul $[2,10^2]$, **k** și **n**, și construiește în memorie un tablou bidimensional cu **n** linii și **n** coloane, astfel încât parcurgând diagonala principală, de sus în jos, să se obțină un șir strict crescător format din primii **n** multipli naturali nenuli ai lui **k** și parcurgând fiecare linie a sa, de la stânga la dreapta, să se obțină câte un șir strict crescător de numere naturale consecutive. Programul afișează pe ecran tabloul obținut, fiecare linie a tabloului pe câte o linie a ecranului, elementele de pe aceeași linie fiind separate prin câte un spațiu.
**Exemplu:** pentru **k=3** și **n=4** se obține tabloul de mai jos

| | | | |
|---|---|---|---|
| 3 | 4 | 5 | 6 |
| 5 | 6 | 7 | 8 |
| 7 | 8 | 9 | 10 |
| 9 | 10 | 11 | 12 |

**(10p.)**

**3.** Numim **pereche asemenea (x,y)** două numere naturale cu cel puțin două cifre, **x** și **y**, cu proprietatea că ultimele două cifre ale lui **x** sunt egale cu ultimele două cifre ale lui **y**, dispuse eventual în altă ordine.
Fișierul **numere.in** conține numere naturale din intervalul $[10,10^5]$: pe prima linie două numere **na** și **nb**, pe a doua linie un șir **A** de **na** numere, iar pe a treia linie un șir **B** de **nb** numere. Numerele aflate pe aceeași linie a fișierului sunt separate prin câte un spațiu.
Se cere să se afișeze pe ecran numărul de perechi asemenea **(x,y)**, cu proprietatea că **x** este un termen al șirului **A**, iar **y** este un termen al șirului **B**. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.
**Exemplu:** dacă fișierul conține numerele
**9 7**
**112 20 42 112 5013 824 10012 55 155**
**402 1024 321 521 57 6542 255**
se afișează pe ecran numărul
**13**
deoarece sunt **13** perechi asemenea: **(112,321)**, **(112,521)**, **(20,402)**, **(42,1024)**, **(42,6542)**, **(112,321)**, **(112,521)**, **(824,1024)**, **(824,6542)**, **(10012,321)**, **(10012,521)**, **(55,255)**, **(155,255)**.
**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 79. Informatica — Subiectul III, exercițiul 1

**Sursă: 2021 · Vara Iunie · V01**

*2021_Informatica_Mate-Info_Sesiune_Vara_Iunie-Iulie_C-Cpp_Subiect_V01.pdf*

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

**1.** Subprogramul **divPrim** are doi parametri:

- **n**, prin care primește un număr natural (**n**$\in$**[2,10$^9$]**);
- **s**, prin care furnizează suma divizorilor primi ai lui **n** care apar la o putere impară în descompunerea în factori primi a acestuia.

Scrieți definiția completă a subprogramului.
**Exemple:** pentru **n=360**, după apel **s=7** (**360=2$^3$·3$^2$·5$^1$**, deci **s=2+5**), iar pentru **n=16**, după apel **s=0**.**(10p.)**

**2.** Scrieți un program C/C++ care citește de la tastatură două numere naturale **n** și **k**, apoi **n** cuvinte, separate prin Enter. Fiecare cuvânt este format din cel mult **10** caractere, numai litere mici ale alfabetului englez, iar numerele citite sunt din intervalul **[1,20]**.

Programul afișează pe ecran, pe linii separate, primele **k** cuvinte dintre cele citite pentru care ultima literă este o vocală, sau **doar** mesajul **nu exista** dacă nu există **k** astfel de cuvinte. Se consideră vocale literele **a**, **e**, **i**, **o**, **u**.
**Exemplu**: dacă se citesc datele alăturate, se afișează pe ecran:
**norii**
**pluteau** **(10p.)**

```
5 2
norii
cumulus
pluteau
pe
cer
```

**3.** Numărul natural **a** se numește **sufix** al numărului natural **b** dacă **a** este egal cu **b** sau dacă **b** se poate obține din **a** prin alipirea la stânga a unor noi cifre.
Fișierul **bac.txt** conţine pe prima linie un număr natural **x** (**x**$\in$**[100,999]**), iar pe a doua linie un şir de cel mult **10$^5$** numere naturale din intervalul **[0,10$^9$]**. Numerele din şir sunt separate prin câte un spaţiu.
Se cere să se afișeze pe ecran ultimii doi termeni ai șirului, aflați pe **poziții consecutive** în acesta, care îl au drept sufix pe numărul **x**. Numerele sunt afișate în ordinea în care apar în șir, separate printr-un spațiu, iar dacă nu există doi astfel de termeni, se afișează pe ecran mesajul **nu exista**. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.

**Exemplu:** dacă fișierul conține numerele alăturate, atunci pe ecran se afișează **3210 15210**

```
210
3445 210 893210 1245 1210 3210 15210 67120 20210 12
```

**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 80. Informatica — Subiectul III, exercițiul 1

**Sursă: 2022 · Model Oficial**

*2022_Informatica_Mate-Info_Model_Oficial_C-Cpp_Subiect.pdf*

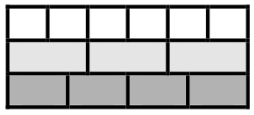
**SUBIECTUL al III-lea** **(30 de puncte)**

**1.** Pentru grădina publică a orașului se dorește proiectarea unui mozaic **de formă dreptunghiulară**, format din plăcuțe dreptunghiulare de ceramică. Furnizorul dispune de trei tipuri de plăcuțe și de suficient de multe plăcuțe din fiecare tip, astfel încât să fie posibilă realizarea oricărui model. Modelul proiectat este constituit din trei benzi suprapuse, fiecare bandă fiind formată din plăcuțe întregi, de același tip, montate unele după altele; oricare două benzi diferite sunt formate din plăcuțe de tipuri diferite.

Subprogramul `mozaic` are trei parametri, `x`, `y` și `z`, prin care primește trei numere naturale din intervalul `[10,200]`, reprezentând lungimea, exprimată în milimetri, a unei plăcuțe de primul tip, de al doilea tip, respectiv de al treilea tip. Subprogramul returnează un număr natural, reprezentând lungimea minimă, exprimată în milimetri, a unui mozaic care să respecte modelul proiectat.

Scrieți definiția completă a subprogramului.

**Exemplu**: pentru `x=30`, `y=40`, `z=20`, subprogramul returnează valoarea `120` (mozaicul are trei benzi, fiecare de lungime `120 mm`: prima formată din `4` plăcuțe de câte `30 mm`, a doua formată din `3` plăcuțe de câte `40 mm`, iar a treia formată din `6` plăcuțe de câte `20 mm`). **(10p.)**



**2.** Un text are cel mult `100` de caractere și este format din cuvinte, numere naturale şi spaţii. Textul conține cel puțin un număr, iar cuvintele sunt formate numai din litere mici ale alfabetului englez. Cuvintele şi numerele sunt separate prin câte un spaţiu, ca în exemplu.
Scrieţi un program C/C++ care citeşte de la tastatură un text de tipul precizat şi afişează pe ecran un număr natural `k`, dacă toate numerele care intră în componența sa au câte `k` cifre, sau mesajul `NU`, în caz contrar.
**Exemplu**: pentru textul `la zoo sunt 100 de cocori si 120 de pasari flamingo`
sau pentru textul `la zoo sunt 100 de cocori`
se afişează `3`
iar pentru textul `la zoo sunt 100 de cocori si 10 pasari flamingo`
se afişează `NU` **(10p.)**

**3.** Numim **secvență par-încadrată** a unui șir de numere naturale un subșir al acestuia, format din termeni aflați pe poziții consecutive în șirul dat, subșir care începe și se termină cu aceeași valoare, pară. Lungimea secvenței este egală cu numărul de termeni ai acesteia.
Fișierul `bac.txt` conține un șir de cel puțin două și cel mult $10^6$ numere naturale din intervalul [`0`, `9`]. Numerele sunt separate prin câte un spațiu, iar în șir există cel puțin doi termeni pari egali.
Se cere să se determine secvențele par-încadrate din acest șir care au lungime maximă și să se afișeze pe prima linie a ecranului lungimea maximă determinată, iar pe următoarea linie, pentru fiecare astfel de secvenţă, valoarea primului său termen. Numerele de pe a doua linie sunt afişate în ordine strict crescătoare, separate prin câte un spaţiu.
Proiectați un algoritm eficient din punctul de vedere al timpului de executare.
**Exemplu:** dacă fişierul `bac.txt` conţine numerele
`5 1 3 2 4 3 3 2 8 9 7 3 4 6 6 0 8`
atunci pe ecran se afișează valorile:
`9`
`4 8`

**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 81. Informatica — Subiectul III, exercițiul 1

**Sursă: 2022 · Simulare Nationala**

*2022_Informatica_Mate-Info_Simulare_Nationala_C-Cpp_Subiect.pdf*

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

**1.** Subprogramul **rest** are patru parametri:
- **x**, **y** și **n**, prin care primește câte un număr natural din intervalul $[1,10^6]$, $x<y<n$;
- **k**, prin care furnizează cea mai mare valoare naturală din intervalul $[1,n]$ pentru care atât restul împărțirii la **x**, cât și restul împărțirii la **y**, sunt egale cu **2**, sau **0** dacă nu există o astfel de valoare.

Scrieți definiția completă a subprogramului.
**Exemplu:** pentru $x=10$, $y=101$ și $n=3000$, subprogramul returnează numărul **2022** (pentru numerele **2**, **1012** și **2022** atât restul împărțirii la **10**, cât și restul împărțirii la **101**, este **2**). **(10p.)**

**2.** Un text, de cel mult **250** de caractere, reprezintă o listă cu date de identificare ale invitaților la o petrecere; fiecare invitat are un prenume și un nume, care apar în listă în această ordine, urmate de simbolul **;** (punct și virgulă), ca în exemplu. Numele și prenumele sunt formate din câte un singur cuvânt, compus din litere mari ale alfabetului englez, și sunt separate printr-un spațiu.
Scrieți un program C/C++ care citește de la tastatură un text de tipul precizat mai sus apoi, de pe rândul următor, un cuvânt, **x**, și afișează pe ecran, separate prin câte un spațiu, **numele** tuturor invitaților care au **prenumele** **x**, ca în exemplu, sau mesajul **NU** dacă nu există astfel de invitați.
**Exemplu:** dacă lista este **DAN MARIS; DANILA PREPELEAC; DAN POPA; EDANA DAN;**
și cuvântul **x** este **DAN** se afișează pe ecran **MARIS POPA** **(10p.)**

**3.** Se citește de la tastatură un număr natural, **n** ($n\in[1,10^9]$), și se cere să se scrie în fișierul text **bac.txt** cel mai mare număr natural **p** cu proprietatea că numărul $45^p$ este divizor al numărului obținut prin evaluarea produsului $1\cdot2\cdot3\cdot\ldots\cdot n$.
Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al memoriei utilizate.
**Exemplu:** dacă **n=14**, fișierul conține numărul **2** ($45^2=2025$ este divizor al lui $1\cdot2\cdot3\cdot\ldots\cdot14=87178291200$)
**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 82. Informatica — Subiectul III, exercițiul 1

**Sursă: 2022 · Speciala C · V04**

*2022_Informatica_Mate-Info_Sesiune_Speciala_C-Cpp_Subiect_V04.pdf*

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

**1.** Subprogramul **patrate** are trei parametri:
- **n**, prin care primeşte un număr natural ($n \in [2, 10^9]$);
- **x** și **y**, prin care furnizează câte un număr natural cu proprietatea că $x^2 \cdot y^2 = n$ și $2 \le x < y$ sau valoarea **0**, prin fiecare dintre aceștia, dacă nu există două astfel de numere. Dacă sunt mai multe astfel de valori, se furnizează cele corespunzătoare unei valori minime a lui **x**.

Scrieţi definiţia completă a subprogramului.
**Exemplu:** pentru **n=400**, după apel, **x=2** şi **y=10**, iar pentru **n=16** sau **n=24**, după apel, **x=0** şi **y=0**. **(10p.)**

**2.** Suprafața unei scene a fost împărțită în zone dispuse ca elementele unui tablou bidimensional cu număr egal de linii și coloane. Fiecare instrument are alocată o zonă și este identificat printr-un număr natural. Dirijorul hotărăște eliminarea tuturor instrumentelor plasate pe diagonala principală a tabloului și rearanjarea celor păstrate, prin deplasarea unora dintre acestea cu câte o poziție spre stânga, astfel încât să nu existe zone intermediare libere, ca în exemplu.
Scrieţi un program C/C++ care citeşte de la tastatură numere naturale din intervalul $[2, 10^2]$, și anume **n** şi elementele unui tablou bidimensional cu **n** linii şi **n** coloane, reprezentând numerele de identificare ale instrumentelor, în ordinea plasării lor inițiale pe scenă. Programul modifică apoi tabloul în memorie corespunzător hotărârii dirijorului și afişează pe ecran tabloul obținut, fiecare linie a tabloului pe câte o linie a ecranului, elementele de pe aceeași linie fiind separate prin câte un spațiu.

**Exemplu:** pentru **n=4** şi tabloul

| | | | |
|---|---|---|---|
| 20 | 30 | 80 | 52 |
| 71 | 93 | 64 | 51 |
| 62 | 81 | 55 | 49 |
| 90 | 65 | 32 | 22 |

se obține tabloul

| | | |
|---|---|---|
| 30 | 80 | 52 |
| 71 | 64 | 51 |
| 62 | 81 | 49 |
| 90 | 65 | 32 |

**(10p.)**

**3.** Numim **secvență progresivă** a unui șir crescător de numere naturale un subșir al acestuia, format din termeni aflați pe poziții consecutive în șirul dat, cu proprietatea că fiecare termen apare în subșir de un număr de ori egal cu valoarea sa. Lungimea secvenței este egală cu numărul de termeni ai acesteia.
Fișierul **bac.txt** conține un șir crescător de cel mult $10^6$ numere naturale din intervalul $[1, 10^6]$, astfel încât orice termen al șirului apare de un număr de ori cel mult egal cu valoarea sa. Numerele sunt separate prin câte un spațiu.
Se cere să se afișeze pe ecran lungimea maximă a unei secvențe progresive din șirul aflat în fișier. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al memoriei utilizate.
**Exemplu:** dacă fişierul conţine numerele 1 2 2 3 4 4 4 4 6 6 6 6 6 6 7 7 7 8 8 8 8 8 8 8 8
atunci pe ecran se afișează valoarea **10**.

**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 83. Informatica — Subiectul III, exercițiul 1

**Sursă: 2022 · Toamna August · V05**

*2022_Informatica_Mate-Info_Sesiune_Toamna_August-Septembrie_C-Cpp_Subiect_V05.pdf*

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

**1.** Subprogramul **schimb** are trei parametri:

- **n** şi **x**, prin care primeşte câte un număr natural (**n**∈**[0,10⁸]**, **x**∈**[1,9]**);
- **p**, prin care primește un număr natural reprezentând poziția unei cifre a numărului **n** (**0≤p**). Pozițiile cifrelor sunt numerotate de la dreapta la stânga, astfel: cifra unităților este pe poziția **0**, cifra zecilor este pe poziția **1** ș.a.m.d.

Subprogramul transformă numărul **n**, înlocuind cifra de pe poziția **p** cu cifra **x**, și furnizează numărul obținut tot prin parametrul **n**. Scrieţi definiţia completă a subprogramului.

**Exemplu:** dacă **n=12587**, **x=6** și **p=3**, după apel, **n=16587**. **(10p.)**

**2.** Se consideră o vocală oarecare a alfabetului englez, notată cu **v**, și o consoană oarecare a alfabetului englez, notată cu **c**. Litera **v** se numeşte **vocală prietenă** a lui **c** dacă în şirul literelor alfabetului englez, ordonat lexicografic, **v** îl precede pe **c**, iar între **v** și **c** nu există nicio vocală. Se consideră vocale literele **a**, **e**, **i**, **o**, **u**.

**Exemplu:** **e** este vocală prietenă pentru consoanele **f**, **g** și **h**, dar nu este vocală prietenă pentru consoanele **d** și **j**.

Un elev vrea să transmită unui prieten o parolă, codificată. Parola este formată dintr-un singur cuvânt de cel mult **50** de caractere, litere mici ale alfabetului englez, cel puțin una fiind consoană. Codificarea se face prin înlocuirea fiecărei consoane cu vocala sa prietenă, ca în exemplu.

Scrieţi un program C/C++ care citeşte de la tastatură un cuvânt, reprezentând o parolă de tipul precizat şi determină, în memorie, forma codificată a acesteia. Programul afişează pe ecran parola codificată obţinută.

**Exemplu**: pentru parola **rame** se afişează **oaie**, iar pentru parola **sport** se afișează **ooooo** **(10p.)**

**3.** Numim **secvență paritară** a unui șir de numere naturale un subșir al acestuia, format din termeni cu aceeași paritate, aflați pe poziții consecutive în șirul dat. Lungimea secvenței este egală cu numărul de termeni ai acesteia.

Fișierul **bac.txt** conține un șir de cel puțin două și cel mult **10⁶** numere naturale din intervalul [**0**, **10⁹**]. Numerele sunt separate prin câte un spațiu, iar în șir există cel puțin doi termeni cu aceeași paritate pe poziții consecutive.

Se cere să se afişeze pe ecran numărul secvențelor paritare de lungime maximă din șirul aflat în fișier, precum și această lungime maximă. Numerele afişate sunt separate printr-un spațiu. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al memoriei utilizate.

**Exemplu:** dacă fişierul conţine numerele

**2 3 5 1 7 9 8 4 4 11 15 17 21 11 6 11 15 17 21 11 6 5 2 6 4 0 16**

atunci pe ecran se afișează valorile **4 5**

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

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

## 84. Informatica — Subiectul III, exercițiul 1

**Sursă: 2022 · Vara Iunie · V01**

*2022_Informatica_Mate-Info_Sesiune_Vara_Iunie-Iulie_C-Cpp_Subiect_V01.pdf*

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

**1.** Subprogramul **secventa** are un singur parametru, **n**, prin care primeşte un număr natural (**n∈[10,10⁹)**) în care nu există secvențe de mai mult de două cifre identice aflate pe poziții consecutive. Subprogramul înlocuiește în **n** fiecare secvență **22** cu câte o secvență **20** și furnizează, prin același parametru, numărul obținut. Dacă nu se înlocuiește nicio secvență, subprogramul furnizează numărul nemodificat.
Scrieţi definiţia completă a subprogramului.

**Exemplu:** dacă **n=202233228**, după apel **n=202033208**. **(10p.)**

**2.** Pentru prevenirea eroziunii la Marea Neagră, autoritățile au hotărât înnisiparea unei plaje. Plaja a fost împărțită în parcele de formă pătrată cu latura de **1** metru, dispuse ca elementele unui tablou bidimensional, reținându-se nivelul fiecărei parcele, măsurat în metri, de la nivelul mării. Înnisiparea se realizează doar pe acele parcele care au nivelul strict mai mic decât al celei mai joase dintre parcelele vecine cu ea, adăugând nisip, astfel încât cele două să aibă același nivel. Două parcele sunt vecine dacă au o latură comună.
Scrieți un program C/C++ care citește de la tastatură două numere naturale din intervalul **[2,10²]**, **m** și **n**, apoi elementele unui tablou bidimensional cu **m** linii şi **n** coloane, numere naturale din intervalul **[1,10]**, reprezentând nivelurile parcelelor, în ordinea dispunerii acestora pe plajă.

Programul afișează pe ecran cantitatea totală de nisip adăugată, măsurată în metri cubi.
**Exemplu**: pentru **m=5**, **n=4** și tabloul alăturat, se vor înnisipa parcelele marcate, pentru a fi aduse la nivelul parcelelor vecine corespunzătoare, și se va afișa valoarea **7** (**1+3+1+2**).

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

**(10p.)**

**3.** Fișierul **bac.txt** conține numere naturale din intervalul **[1,10⁹]**, astfel: pe prima linie două numere, **x** și **y** (**x<y**), iar pe a doua linie un șir de cel mult **10⁶** numere, ordonate crescător. Numerele aflate pe aceeași linie a fișierului sunt separate prin câte un spațiu.
Se cere să se afişeze pe ecran numărul de valori distincte din șirul aflat pe a doua linie a fișierului care aparțin intervalului **[x,y]**. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al memoriei utilizate.

**Exemplu:** dacă fişierul are conținutul alăturat, pe ecran se afișează **6**

```
2 9
1 1 1 2 2 3 5 5 5 5 6 6 7 8 10 10 12 15 21 21
```

**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 85. Informatica — Subiectul III, exercițiul 1

**Sursă: 2023 · Model Oficial**

*2023_Informatica_Mate-Info_Model_Oficial_C-Cpp_Subiect.pdf*

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

**1.** Subprogramul **DoiTrei** are un parametru, **n**, prin care primește un număr natural ($n \in [0, 10^9]$). Subprogramul returnează valoarea **1** dacă toate cifrele lui **n** sunt din mulțimea **{2,3}** sau valoarea **0** în caz contrar. Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **n=22323** sau **n=3**, atunci subprogramul returnează **1**, iar dacă **n=2023** atunci subprogramul returnează **0**. **(10p.)**

**2.** Suprafața unui loc de joacă pentru copii este împărțită în **n·n** zone de formă pătrată dispuse pe **n** rânduri și **n** coloane, numerotate de la **1** la **n**, în fiecare zonă fiind plasat un tobogan, un leagăn, un balansoar sau un carusel. Zonelor li se atribuie câte un cod, în funcție de destinație: codul **0** pentru tobogan, codul **1** pentru leagăn, codul **2** pentru balansoar și codul **3** pentru carusel.
Scrieți un program C/C++ care citește de la tastatură un număr natural, **n** ($n \in [1, 10^2]$), și cele **n·n** elemente ale unui tablou bidimensional, reprezentând codurile corespunzătoare zonelor din locul de joacă, în ordinea parcurgerii lor rând după rând, și a fiecărui rând de la stânga la dreapta.
Programul determină zonele destinate plasării câte unui carusel, care NU au nicio latură comună cu o zonă cu tobogan și, pentru fiecare dintre acestea, afișează pe ecran numărul de ordine al rândului pe care se află. Numerele afișate sunt separate prin câte un spațiu, iar dacă nu există nicio astfel de zonă, se afișează pe ecran mesajul **nu exista**.
**Exemplu:** pentru **n=6** și tabloul alăturat se afișează pe ecran, nu neapărat în această ordine, numerele: **2 3 3 6**. **(10p.)**

| | | | | | |
|---|---|---|---|---|---|
| 2 | 0 | 0 | 2 | 1 | 3 |
| 3 | 1 | 3 | 1 | 2 | 0 |
| 2 | 1 | 3 | 3 | 2 | 0 |
| 0 | 2 | 1 | 3 | 1 | 1 |
| 3 | 1 | 2 | 0 | 0 | 2 |
| 0 | 0 | 0 | 2 | 1 | 3 |

**3.** O pereche de numere naturale **(x,y)** se numește **p-ordonată** dacă **x** și **y** au paritate diferită, cu **x>y**, sau dacă **x** și **y** au aceeași paritate.
**Exemplu:** **(7,4)**, **(8,7)**, **(7,9)** sunt perechi p-ordonate, dar **(7,8)**, **(8,9)** nu sunt perechi p-ordonate.
Un șir, format din **2·n** numere naturale, se numește **p-ordonat** dacă perechile de forma **(x,y)** sunt p-ordonate, pentru orice **x**, dintre primii **n** termeni ai șirului, și orice **y**, dintre ultimii **n** termeni ai șirului.
Fișierul **bac.txt** conține numere naturale din intervalul **[0,10⁶]**: pe prima linie un număr nenul, **n**, iar pe a doua linie un șir de **2·n** numere, separate prin câte un spațiu. Se cere să se afișeze pe ecran mesajul **DA**, în cazul în care șirul aflat în fișier este p-ordonat, sau mesajul **NU**, în caz contrar. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al memoriei utilizate.
**Exemplu:** dacă fișierul are unul dintre conținuturile de mai jos, se afișează pe ecran mesajul **DA**.

```
5
22 37 21 8 15 2 3 14 4 5
```

```
5
22 36 20 8 14 44 42 4 2 5
```

**a)** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b)** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 86. Informatica — Subiectul III, exercițiul 1

**Sursă: 2023 · Simulare Nationala**

*2023_Informatica_Mate-Info_Simulare_Nationala_C-Cpp_Subiect.pdf*

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

**1.** Subprogramul **NrImp** are trei parametri:
- **x** și **y**, prin care primeşte câte un număr natural ($2 \le x < y \le 10^9$)
- **nr**, prin care furnizează numărul valorilor naturale din intervalul **[x,y]** cu trei divizori pozitivi impari.

Scrieţi definiţia completă a subprogramului.
**Exemplu:** dacă **x=4** și **y=50**, după apel **nr=6** (pentru valorile **9**, **18**, **25**, **36**, **49**, **50**). **(10p.)**

**2.** Într-un text de cel mult **100** de caractere cuvintele sunt separate prin câte un spațiu și sunt formate din litere mari ale alfabetului englez, iar dacă sunt scrise prescurtat sunt urmate de caracterul **.** (punct). Textul reprezintă denumirea științifică a unei păsări și doar cuvintele din mulțimea {**FAMILIA**, **GENUL**, **SPECIA**}, specifice sistemului de clasificare a organismelor, sunt mereu prescurtate, prin eliminarea ultimelor lor litere.
Scrieţi un program C/C++ care citeşte de la tastatură un text de tipul precizat și construiește în memorie, apoi afișează pe ecran denumirea științifică, în care pentru cuvintele specifice sistemului de clasificare a organismelor se păstrează doar primele trei litere, scrise cu litere mici, și urmate de punct, ca în exemplu.
**Exemplu:** pentru textul **FAMIL. PHASIANIDAE GEN. MELEAGRIS SP. GALLOPAVO**
sau pentru textul **FAM. PHASIANIDAE G. MELEAGRIS SPECI. GALLOPAVO**
se obține **fam. PHASIANIDAE gen. MELEAGRIS spe. GALLOPAVO** **(10p.)**

**3.** Pentru a studia un metal, s-a urmărit comportamentul său într-o succesiune de pași, la fiecare pas metalul fiind supus unei anumite temperaturi. Pașii sunt numerotați cu valori naturale consecutive, începând de la **1**. Un pas se numește **reprezentativ** dacă la niciunul dintre pașii anteriori nu este utilizată o temperatură strict mai mare decât la acest pas. Dacă există o secvență de pași consecutivi la care se utilizează aceeași temperatură, se consideră reprezentativ doar primul pas din secvență.
Fișierul **bac.txt** conține cel mult $10^6$ numere naturale din intervalul $[0, 10^4]$, separate prin câte un spațiu, reprezentând temperaturile la care este supus metalul, în ordinea pașilor corespunzători. Se cere să se afișeze pe ecran, separați prin câte un spațiu, pașii reprezentativi pentru datele din fișier. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al spațiului de memorie utilizat.
**Exemplu:** dacă fișierul conține numerele **7 4 9 10 10 10 3 9 2 10 10 8 2 30**
se afișează pe ecran **1 3 4 10 14**
**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 87. Informatica — Subiectul III, exercițiul 1

**Sursă: 2023 · Speciala C · V06**

*2023_Informatica_Mate-Info_Sesiune_Speciala_C-Cpp_Subiect_V06.pdf*

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

**1.** Subprogramul **Putere** are trei parametri:

- **n**, prin care primește un număr natural ($n \in [2, 10^9]$);
- **x** și **p**, prin care furnizează două numere naturale cu proprietatea că $n = x^p$, iar **x** este cel mai mic număr cu această proprietate.

Scrieți definiția completă a subprogramului.

**Exemplu:** dacă **n=16** atunci, după apel, **x=2** și **p=4**, dacă **n=216** atunci, după apel, **x=6** și **p=3**, iar dacă **n=12** atunci, după apel, **x=12** și **p=1**. **(10p.)**

**2.** La o expoziție sunt prezentate mai multe tipuri de flori, fiecare având un anumit cod. Zona de expunere este organizată pe **n** rânduri, fiecare având câte **n** parcele de dimensiune egală. Pe primul rând sunt expuse **n** tipuri de flori distincte, fiecare parcelă conținând flori de un anumit tip. Celelalte parcele sunt ocupate astfel:

- toate parcelele aflate în dreapta diagonalei secundare a zonei de expunere sunt ocupate de gazon;
- fiecare parcelă, aflată pe diagonala secundară a zonei de expunere sau în stânga acestei diagonale, conține flori cu un cod de cel mult două cifre, obținut din ultimele două cifre ale sumei codurilor florilor aflate în două parcele de pe rândul anterior, și anume cea de pe aceeași coloană, respectiv cea de pe coloana din dreapta sa; dacă prima dintre cifrele codului obținut este **0**, ea este ignorată, ca în exemplu.

Scrieți un program C/C++ care citește de la tastatură un număr natural, **n** ($n \in [2, 20]$), apoi **n** numere naturale distincte din intervalul $[0, 10^2)$, reprezentând, în această ordine, de la stânga la dreapta, codurile florilor din parcelele aflate pe primul rând al zonei de expunere; gazonul are codul **-1**. Programul construiește în memorie un tablou bidimensional cu **n** linii și **n** coloane, în care fiecare linie corespunde unui rând de parcele și fiecare element al liniei corespunde unei parcele, în ordinea acestora, și memorează codul plantelor (flori sau gazon) plasate în acea parcelă.

Programul afișează pe ecran tabloul obținut, fiecare linie a tabloului pe câte o linie a ecranului, elementele de pe aceeași linie fiind separate prin câte un spațiu.

**Exemplu:** dacă se citesc numerele **n=5**, apoi **90**, **10**, **4**, **27**, **30** se obține tabloul alăturat. **(10p.)**

| | | | | |
|---|---|---|---|---|
| 90 | 10 | 4 | 27 | 30 |
| 0 | 14 | 31 | 57 | -1 |
| 14 | 45 | 88 | -1 | -1 |
| 59 | 33 | -1 | -1 | -1 |
| 92 | -1 | -1 | -1 | -1 |

**3.** Intervalul **[x,y]** se numește **p**-**interval** pentru un șir de valori întregi, dacă oricare dintre primii **p** termeni ai șirului aparține intervalului, iar numărul de valori întregi distincte din interval este minim.

**Exemplu:** pentru șirul **2**, **7**, **-1**, **8**, **3**, **10** există **[2,2]** ca **1**-interval, **[-1,8]** ca **4**-interval și **5**-interval etc.

Fișierul **bac.in** conține un șir de cel mult $10^6$ numere întregi din intervalul $[-10^9, 10^9]$, separate prin câte un spațiu.

Se cere să se afișeze pe ecran cea mai mică și cea mai mare valoare a lui **p** ($p \geq 2$) cu proprietatea că **(p-1)**-interval este identic cu **p**-interval pentru șirul aflat în fișier. Valorile afișate pot fi egale, iar dacă nu există nicio astfel de valoare, pe ecran se afișează mesajul **nu exista**. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al memoriei utilizate.

**Exemplu:** dacă fișierul conține numerele **2 7 1 8 3 10 6 -3 -2 13**, se afișează pe ecran **5 9** (intervale conform cerinței se obțin pentru valorile **5**, **7** și **9** ale lui **p**),

iar dacă fișierul conține numerele **2 7 1 0 8 10 -3 13**, se afișează pe ecran **nu exista**

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

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

## 88. Informatica — Subiectul III, exercițiul 1

**Sursă: 2023 · Toamna August · V07**

*2023_Informatica_Mate-Info_Sesiune_Toamna_August-Septembrie_C-Cpp_Subiect_V07.pdf*

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

**1.** Subprogramul **DNPI** are un singur parametru, **n**, prin care primește un număr natural ($n \in [1, 10^9]$), și afișează pe ecran, separați prin câte un spațiu, toți divizorii pozitivi impari ai lui **n** care **NU** sunt primi. Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **n=90**, se afișează pe ecran, nu neapărat în această ordine, numerele
**1 9 15 45** **(10p.)**

**2.** Un joc folosește o tablă dreptunghiulară, pe care sunt reprezentate celule de dimensiune egală, dispuse pe **m** linii și pe **n** coloane. În fiecare celulă este înscris un număr natural.
Numim **pătrat de valoare p** patru celule ale tablei, situate pe două linii consecutive și pe două coloane consecutive, cu proprietatea că suma valorilor înscrise în aceste celule este egală cu **p**. Scopul jocului este determinarea unui pătrat de valoare maximă.
Scrieți un program C/C++ care citește de la tastatură două numere naturale din intervalul **[2,20]**, **m** și **n**, apoi elementele unui tablou bidimensional cu **m** linii și **n** coloane, numere naturale din intervalul $[0, 10^4]$, reprezentând numerele înscrise pe tabla de joc, în ordinea dispunerii celulelor corespunzătoare. Programul determină un pătrat de valoare maximă al tablei și afișează pe ecran această valoare.

```
1  1  2 30
3 10  1  2
1 13 12  1
1  2  3  1
7  1 20  1
```

**Exemplu:** pentru **m=5**, **n=4** și tabloul alăturat, se afișează pe ecran valoarea **36**, corespunzătoare pătratului evidențiat în figură. **(10p.)**

**3.** Fișierul **date.in** conține pe prima linie două numere naturale din intervalul $[1, 10^6]$, **m** și **n**, iar pe următoarele două linii numere naturale din intervalul $[0, 10^2)$: pe a doua linie un șir **A**, de **m** numere, iar pe a treia linie un șir **B**, de **n** numere. Numerele aflate pe aceeași linie sunt separate prin câte un spațiu. Se cere să se afișeze pe ecran numărul maxim de perechi de forma **(pa,pb)** ($pa \in [1,m]$, $pb \in [1,n]$), cu proprietatea că termenul de pe poziția **pa** din șirul **A** are aceeași valoare cu termenul de pe poziția **pb** din șirul **B** și că fiecare poziție, corespunzătoare șirului **A**, respectiv șirului **B**, apare în cel mult o pereche, ca în exemplu. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.
**Exemplu**: dacă fișierul conține numerele alăturate, se afișează pe ecran **6** (de exemplu, pentru perechile **(1,1)**, **(2,9)**, **(4,2)**, **(5,5)**, **(6,6)**, **(7,7)** sau pentru perechile **(1,2)**, **(2,9)**, **(4,1)**, **(5,7)**, **(6,8)**, **(8,5)**).

```
8 9
1 0 4 1 5 3 5 5
1 1 1 7 5 3 5 3 0
```

**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 89. Informatica — Subiectul III, exercițiul 1

**Sursă: 2023 · Vara Iunie · V05**

*2023_Informatica_Mate-Info_Sesiune_Vara_Iunie-Iulie_C-Cpp_Subiect_V05.pdf*

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

**1.** Un număr natural nenul, **n**, se numește număr **abundent** dacă **S(n)/n>S(k)/k**, pentru orice număr natural nenul **k** (**k≤n−1**), unde s-a notat cu **S(i)** suma divizorilor pozitivi ai numărului natural nenul **i**.
Subprogramul **abundent** are un singur parametru, **n**, prin care primește un număr natural (**n∈[2,10⁶]**). Subprogramul returnează valoarea **1**, dacă **n** este un număr abundent, sau valoarea **0**, în caz contrar. Scrieți definiția completă a subprogramului.
**Exemplu:** pentru **n=6**, subprogramul returnează valoarea **1** (**S(6)/6=2**, iar cel mai mare raport obținut pentru valori strict mai mici decât **6** este **S(4)/4=1.75**), iar pentru **n=7** sau **n=8**, subprogramul returnează valoarea **0** (**S(7)/7=1.14**, **S(8)/8=1.87**). **(10p.)**

**2.** Pentru a identifica punctele în care se concentrează apa în albia unui râu în cazul secetei, se determină talvegul acesteia – linia care unește punctele cele mai adânci ale albiei. În acest scop s-au stabilit **ns** secțiuni transversale pe cursul apei, numerotate începând de la **1**, și în cadrul fiecărei secțiuni s-a măsurat adâncimea apei în **np** puncte, numerotate începând de la **1**. Din fiecare secțiune, în ordine, se include în talveg cel mai adânc punct al acesteia, iar dacă în secțiune sunt mai multe puncte aflate la aceeași adâncime, maximă, se va lua în considerare doar primul dintre ele, ca în exemplu.
Scrieți un program C/C++ care citește de la tastatură două numere naturale, **ns** și **np** (**ns∈[1,10³]**, **np∈[1,50]**), și cele **ns·np** elemente ale unui tablou bidimensional, valori naturale din intervalul **[0,10²]**. Fiecare linie a tabloului corespunde câte unei secțiuni, în ordinea numerotării acestora, iar valorile memorate pe linie reprezintă adâncimile celor **np** puncte stabilite pentru acea secțiune, în ordinea numerotării lor. Programul afișează pe ecran, pentru fiecare secțiune, o pereche formată din numărul de ordine al secțiunii și numărul de ordine al punctului său care s-a inclus în talveg. Numerele din fiecare pereche sunt afișate separate prin câte un caracter **:** (două puncte), iar fiecare pereche este urmată de un spațiu.

| | | | |
|---|---|---|---|
| 2 | 4 | 5 | 3 |
| 2 | 6 | 6 | 3 |
| 1 | 5 | 2 | 5 |
| 1 | 3 | 3 | 3 |
| 3 | 4 | 3 | 5 |
| 0 | 1 | 2 | 1 |

**Exemplu:** pentru **ns=6**, **np=4** și tabloul alăturat, se afișează pe ecran valorile:
**1:3 2:2 3:2 4:2 5:4 6:3** **(10p.)**

**3.** Un număr natural **x** este numit prefix al unui număr natural **y** dacă se obține din acesta prin eliminarea a cel puțin unei cifre de la dreapta sa, și este numit sufix al lui **y** dacă se obține din acesta prin eliminarea a cel puțin unei cifre de la stânga sa.
**Exemplu:** **15** este prefix pentru **154** sau **1521**, este sufix pentru **3415** sau **5115**, dar nu este nici prefix, nici sufix pentru **15**.
Fișierul **bac.txt** conține maximum **10⁶** numere naturale din intervalul **[10,10⁴)**, separate prin câte un spațiu. Se cere să se afișeze pe ecran numărul valorilor de două cifre care apar de același număr de ori ca sufix, respectiv ca prefix al numerelor din șirul aflat în fișier. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.
**Exemplu:** dacă fișierul are conținutul
**342 1684 2134 5434 111 98 98 3405 3412 7016 8634 1010 102 310**
se afișează pe ecran: **4** (pentru valorile **10**, **11**, **16**, **34**).
**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 90. Informatica — Subiectul III, exercițiul 1

**Sursă: 2024 · Model Oficial**

*2024_Informatica_Mate-Info_Model_Oficial_C-Cpp_Subiect.pdf*

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

**1.** Subprogramul **produs** are doi parametri, **a** și **b**, prin care primește câte un număr natural din intervalul $[1,10^3]$. Subprogramul returnează produsul divizorilor naturali comuni lui **a** și **b**.
Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **a=20** și **b=12**, atunci subprogramul returnează valoarea **8** ($1 \cdot 2 \cdot 4 = 8$). **(10p.)**

**2.** Un text are cel mult **100** de caractere, iar cuvintele sale sunt formate numai din litere mici ale alfabetului englez, sunt distincte și sunt separate prin câte un spațiu. Scrieți un program C/C++ care citește de la tastatură un număr natural **n** ($n \in [1,10^2]$), apoi un text de tipul precizat mai sus, și afișează pe ecran cuvinte ale acestuia, pe două linii separate, astfel încât prima linie să conțină mulțimea cuvintelor care au mai puțin de **n** litere, iar a doua linie să conțină mulțimea cuvintelor care au mai mult de **n** litere. Cuvintele de pe fiecare linie sunt afișate într-o ordine oarecare, iar dacă una dintre cele două mulțimi este vidă, se afișează pe ecran doar mesajul **nu exista**.

**Exemplu**: pentru **n=3** și textul **era o apa rece si cu gust bun** se poate afișa pe ecran textul alăturat. **(10p.)**

```
o si cu
rece gust
```

**3.** La un concurs se acordă premiile I, al II-lea și al III-lea. Fiecare premiant este recompensat cu câte o carte, care are un preț egal pentru toți cei cu același premiu. Prețurile cărților alese pentru premiile I, al II-lea și al III-lea sunt stabilite astfel încât să fie în ordine strict descrescătoare, iar pentru fiecare premiu să se ia în considerare cel mai mare preț pentru care există suficiente cărți propuse, în condițiile precizate.
Fișierul **bac.txt** conține pe prima linie trei numere naturale din intervalul **[1,20]**, **n1**, **n2** și **n3**, reprezentând numărul concurenților care primesc premiile I, al II-lea, respectiv al III-lea, iar pe a doua linie un șir de cel mult $10^6$ numere naturale din intervalul $[10,10^3]$, separate prin câte un spațiu, fiecare număr reprezentând prețul unei cărți propuse pentru premiere.
Se cere să se afișeze pe ecran, separate prin câte un spațiu, în ordine strict descrescătoare, prețurile cărților alese, corespunzătoare celor trei premii, iar dacă nu există trei astfel de prețuri, se afișează mesajul **nu exista**. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.
**Exemplu:** dacă fișierul conține valorile
**3 2 4**
**500 100 25 100 200 100 20 10 200 100 75 52 52 15 52 20 20 10 30 20 15 100**
se afișează pe ecran, în această ordine, numerele **100 52 20**.
**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 91. Informatica — Subiectul III, exercițiul 1

**Sursă: 2024 · Simulare Nationala**

*2024_Informatica_Mate-Info_Simulare_Nationala_C-Cpp_Subiect.pdf*

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

**1.** Subprogramul **Impare** are un parametru, **n**, prin care primește un număr natural ($n \in [1, 10^4]$). Subprogramul returnează numărul obținut prin duplicarea fiecărei cifre impare a lui **n** sau **-1** dacă acesta nu are nicio cifră impară. Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **n=3361**, subprogramul returnează **3333611**. **(10p.)**

**2.** Un **șablon** este un text în care cuvintele sunt separate prin câte un spațiu și sunt formate fie numai din litere mici și mari ale alfabetului englez, fie numai din caractere *, în ultimul caz numindu-se **cuvinte generice**. Lungimea unui cuvânt este egală cu numărul de caractere care îl compun.
Un computer generează o frază pe baza unui astfel de șablon, prin înlocuirea fiecărui cuvânt generic cu unul dintre cuvintele de aceeași lungime, preluat dintr-o listă dată.
Scrieți un program C/C++ care citește de la tastatură un număr natural, **n** ($n \in [1, 100]$), și o listă de **n** cuvinte, urmată de un șablon de tipul precizat. Fiecare cuvânt din listă este format din maximum **10** litere mici și mari ale alfabetului englez și la citire este introdus singur pe linie. Șablonul conține maximum **100** de caractere. Programul obține în memorie și apoi afișează pe ecran una dintre frazele care pot fi generate pe baza șablonului și a listei citite sau mesajul **imposibil** dacă nu se poate genera o astfel de frază.

**Exemplu**: dacă **n=6**, iar lista de cuvinte este cea alăturată,

| | |
|---|---|
| pentru șablonul | `Era o vreme ******* si ***** din belsug *****` |
| se generează fraza | `Era o vreme placuta si soare din belsug soare` |
| sau fraza | `Era o vreme calduta si soare din belsug acasa` |
| etc., iar pentru șablonul | `*** o vreme ******* si ************` |
| se afișează mesajul **imposibil** | **(10p.)** |

```
rece
placuta
acasa
calduta
innorata
soare
```

**3.** Un șir se numește de tip **api** dacă numărul de apariții ale fiecărui termen este mai mic sau egal cu acel termen și are o paritate egală cu a acestuia.
Fișierul **bac.in** conține un șir de cel mult $10^6$ numere naturale din intervalul $[1, 10^3]$, separate prin câte un spațiu. Se cere să se afișeze pe ecran mesajul **DA**, dacă șirul este de tip api, sau mesajul **NU** în caz contrar. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.
**Exemplu:** dacă fișierul conține numerele **6 27 2 6 27 6 6 14 14 2 27** se afișează pe ecran **DA** (termenul par **6** apare de **4** ori, **4** fiind tot număr par și $4 \leq 6$, termenii pari **2** și **14** apar de câte **2** ori, **2** fiind tot număr par și $2 \leq 2$, respectiv $2 \leq 14$, iar termenul impar **27** apare de **3** ori, **3** fiind tot număr impar și $3 \leq 27$).
**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 92. Informatica — Subiectul III, exercițiul 1

**Sursă: 2024 · Speciala C · V04**

*2024_Informatica_Mate-Info_Sesiune_Speciala_C-Cpp_Subiect_V04.pdf*

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

**1.** La un laborator sunt studiate aglomerările de fulgi de nea formate din câte **nouă** cristale de **patru** tipuri diferite date (notate cu `1`, `2`, `3` sau `4`), astfel încât din fiecare tip să existe cel puțin câte un cristal. O astfel de aglomerare de fulgi a fost reprezentată printr-un număr natural, în care fiecare cifră reprezintă tipul unui cristal.
Subprogramul `fulg` are un parametru, `n`, prin care primește un număr natural ($n \in [0, 10^9)$). Subprogramul returnează valoarea `1`, dacă prin `n` este reprezentată o aglomerare de fulgi de nea dintre cele studiate, sau `0` în caz contrar. Scrieți în C/C++ definiția completă a subprogramului.
**Exemplu:** dacă `n=112243413` subprogramul returnează `1`, iar dacă `n=12314` sau `n=112253513` sau `n=112243457` sau `n=111122223`, subprogramul returnează `0`. **(10p.)**

**2.** Fiind date două numere naturale `a` și `b`, numim valoare **generată** de `a` și `b` un număr obținut din `a` prin alipirea la stânga sau la dreapta sa a cifrelor lui `b`, în ordinea în care apar în acesta.
**Exemplu:** dacă `a=123` și `b=45`, se pot obține două numere generate de acestea: `12345` și `45123`.
Scrieţi un program C/C++ care citeşte de la tastatură două numere naturale din intervalul `[2,50]`, `m` și `n`, şi construiește în memorie un tablou bidimensional cu `m` linii, numerotate de la `1` la `m`, și `n` coloane, numerotate de la `1` la `n`, în care fiecare element este egal cu cea mai mică valoare generată de numerele de ordine ale liniei, respectiv coloanei pe care se află.
Programul afişează pe ecran tabloul obţinut, fiecare linie a tabloului pe câte o linie a ecranului, elementele fiecărei linii fiind separate prin câte un spaţiu.
**Exemplu:** pentru `m=5` și `n=4` se obține tabloul alăturat.
**(10p.)**

```
11  12  13  14
12  22  23  24
13  23  33  34
14  24  34  44
15  25  35  45
```

**3.** Fişierul `numere.in` conţine un şir de cel mult $10^6$ numere naturale din intervalul `[0,99]`. Numerele din fişier sunt separate prin câte un spaţiu.
Se cere să se determine primul şi ultimul număr din şir care conţin cea mai mare cifră ce apare în scrierea numerelor din fișier. Numerele determinate se afişează pe ecran, în ordinea apariţiei lor în şir, separate printr-un spaţiu. Dacă nu există două astfel de numere pe poziții distincte, se afişează pe ecran mesajul `nu exista`.
Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al spațiului de memorie utilizat.
**Exemplu:** dacă fişierul conţine numerele `34 5 38 30 87 70 11 8 82 25` se afişează pe ecran `38 82`, dacă fişierul conţine numerele `34 5 38 30 87 70 11 8 38 25` se afişează pe ecran `38 38`, iar dacă fişierul conţine numerele `34 5 38 30` se afişează pe ecran `nu exista`.

**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 93. Informatica — Subiectul III, exercițiul 1

**Sursă: 2024 · Toamna August · V08**

*2024_Informatica_Mate-Info_Sesiune_Toamna_August-Septembrie_C-Cpp_Subiect_V08.pdf*

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

**1.** Un număr natural nenul, **n**, se numește **moderat** dacă este egal cu produsul a două numere prime, iar acestea sunt consecutive în șirul numerelor prime (**2**, **3**, **5**, **7**, **11**, **13**, **17**...).
Subprogramul **moderat** are un singur parametru, **n**, prin care primește un număr natural ($n \in [1,10^5]$). Subprogramul returnează valoarea **1**, dacă **n** este un număr moderat, sau valoarea **0**, în caz contrar.
Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **n=35**, subprogramul returnează **1** (**35=5·7**), iar dacă **n=18** sau **n=55** sau **n=4**, subprogramul returnează **0**. **(10p.)**

**2.** O grădină este împărţită în parcele pătrate identice, dispuse pe rânduri succesive. Toate parcelele de pe un rând conțin același tip de plante (doar flori sau doar gazon) și oricare două rânduri alăturate conțin tipuri diferite de plante, **pe primul rând fiind gazon**. Pentru fiecare parcelă se memorează înălţimea acesteia, egală cu înălțimea medie a plantelor conținute, exprimată în decimetri. Pentru a pune în valoare florile, fiecare parcelă cu flori trebuie să fie mai înaltă cu cel puțin un decimetru decât oricare dintre parcelele cu gazon de pe rândul vecin anterior. S-a hotărât tunderea parcelelor cu gazon prea înalte, astfel încât ele să fie aduse la înălțimea maximă permisă.
Scrieți un program C/C++ care citește de la tastatură două numere naturale din intervalul $[2,10^2]$, **m** și **n**, apoi elementele unui tablou bidimensional cu **m** linii şi **n** coloane, numere naturale din intervalul **[2,10]**, reprezentând înălţimile parcelelor, în ordinea dispunerii acestora în grădină, rând după rând, și pe fiecare rând de la stânga la dreapta. Programul modifică apoi tabloul în memorie corespunzător situației terenului după tunderea gazonului și afişează pe ecran tabloul obținut, fiecare linie a tabloului pe câte o linie a ecranului, elementele de pe aceeași linie fiind separate prin câte un spațiu.
**Exemplu**:
pentru **m=4**, **n=6** și tabloul

| | | | | | |
|---|---|---|---|---|---|
| 6 | 2 | 3 | 7 | 9 | 4 |
| 7 | 8 | 5 | 9 | 8 | 5 |
| 8 | 7 | 2 | 6 | 2 | 9 |
| 7 | 2 | 5 | 9 | 3 | 8 |

se obține tabloul

| | | | | | |
|---|---|---|---|---|---|
| 4 | 2 | 3 | 4 | 4 | 4 |
| 7 | 8 | 5 | 9 | 8 | 5 |
| 1 | 1 | 1 | 1 | 1 | 1 |
| 7 | 2 | 5 | 9 | 3 | 8 |

**(10p.)**

**3.** Șirul **0**, **0**, **1**, **4**, **13**, **38**, **105**, **280**, **729** . . . .
este definit astfel: $f_1=f_2=0$, $f_3=1$, $f_n=4\cdot f_{n-1}-3\cdot f_{n-2}-2\cdot f_{n-3}$ (unde **n** este un număr natural $n\geq 4$).
Se citesc de la tastatură trei numere naturale **x**, **y** și **z** ($x\leq y<z\leq 10^9$), valorile a trei termeni aflați pe poziții consecutive în şirul dat, şi se cere să se scrie în fişierul **bac.txt**, în ordine descrescătoare, separați prin câte un spațiu, toţi termenii şirului care sunt mai mici sau egali cu **z**. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al spațiului de memorie utilizat.
**Exemplu:** dacă se citesc numerele **38 105 280** fişierul conţine numerele **280 105 38 13 4 1 0 0**

**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 94. Informatica — Subiectul III, exercițiul 1

**Sursă: 2024 · Vara Iunie · V03**

*2024_Informatica_Mate-Info_Sesiune_Vara_Iunie-Iulie_C-Cpp_Subiect_V03.pdf*

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

**1.** Un număr natural se numește **major impar** dacă suma divizorilor săi proprii impari este strict mai mare decât suma divizorilor săi proprii pari. Divizorii proprii ai unui număr sunt divizorii săi naturali diferiți de **1** și de el însuși.
**Exemplu:** **18** este număr major impar (divizorii săi proprii pari sunt **2**, **6**, cei impari **3**, **9**, iar **3+9>2+6**).
Subprogramul **majImp** are doi parametri, **a** și **b**, prin care primește câte un număr natural ($2 \le a \le b \le 10^4$). Subprogramul returnează cel mai mic număr major impar din intervalul **[a,b]**, sau valoarea **0**, dacă în interval nu există un astfel de număr. Scrieţi în C/C++ definiţia completă a subprogramului.
**Exemplu:** dacă **a=16**, **b=30**, atunci subprogramul returnează **18**. **(10p.)**

**2.** Într-un text, de cel mult **100** de caractere, cuvintele sunt formate din litere ale alfabetului englez și sunt separate prin câte un spațiu. Textul are cel puțin două cuvinte.
Scrieți un program C/C++ care citește de la tastatură un text de tipul precizat mai sus și afișează pe ecran mesajul **DA** și un număr natural **n**, separate printr-un spațiu, dacă toate cuvintele din text au câte **n** litere, sau mesajul **NU** în cazul în care nu toate cuvintele au același număr de litere.
**Exemplu**: dacă textul citit este **Ana are cel mai bun mar** se afișează pe ecran **DA 3**
iar dacă textul citit este **Ana are cel mai dulce mar** se afișează pe ecran **NU** **(10p.)**

**3.** De-a lungul unui traseu montan este utilizată o succesiune de marcaje turistice, care trebuie urmate în acea ordine. Pentru fiecare marcaj se cunoaște cota (înălțimea, măsurată în metri) la care este plasat. Numim **scară** într-un traseu o secvență de marcaje aflate pe poziții consecutive în cadrul traseului, care au drept cote numere consecutive, ordonate strict crescător. O scară este formată din cel puțin două marcaje, iar lungimea acesteia este egală cu numărul de marcaje care o compun.
Fișierul **bac.txt** conține un șir de cel mult $10^6$ numere naturale din intervalul $[10, 10^4]$, separate prin câte un spațiu, reprezentând cotele marcajelor turistice din cadrul unui traseu, în ordinea în care se succed în acesta. Se cere să se afișeze pe ecran, separate prin câte un spațiu, în ordine strict crescătoare, cotele corespunzătoare marcajelor unei scări de lungime maximă pe acest traseu. Dacă în cadrul traseului există mai multe astfel de scări, se afișează doar cotele corespunzătoare marcajelor uneia dintre ele, iar dacă nu există nicio scară, pe ecran se afișează mesajul **nu exista**. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al spațiului de memorie utilizat.
**Exemplu:** dacă fișierul conține numerele **500 600 601 405 569 570 700 701 625 626 627 520**
atunci pe ecran se afișează **625 626 627**

**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 95. Informatica — Subiectul III, exercițiul 1

**Sursă: 2025 · Model Oficial**

*2025_Informatica_Mate-Info_Model_Oficial_C-Cpp_Subiect.pdf*

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

**1.** Două numere se numesc **oglindite** dacă fiecare se obține din celălalt, prin parcurgerea cifrelor acestuia de la dreapta la stânga. Două numere se numesc **par-oglindite** dacă numerele obținute din acestea, prin îndepărtarea tuturor cifrelor lor impare sau nule, sunt oglindite.
Subprogramul **pao** are trei parametri:
- **x** și **y**, prin care primește câte un număr natural din intervalul $[0,10^9]$;
- **rez**, prin care furnizează valoarea **1**, dacă **x** și **y** sunt par-oglindite, sau valoarea **0**, în caz contrar.

Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **x=814** și **y=7003485**, sau **x=14** și **y=70345**, după apel **rez=1**,
iar dacă **x=814** și **y=7003465**, sau **x=814** și **y=7003845**, sau **x=15** și **y=510**, după apel **rez=0**. **(10p.)**

## 96. Informatica — Subiectul III, exercițiul 1

**Sursă: 2025 · Rezerva Vara Iunie · V07**

*2025_Informatica_Mate-Info_Rezerva_Vara_Iunie-Iulie_C-Cpp_Subiect_V07.pdf*

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

**1.** Subprogramul **`consecutiv`** are doi parametri:

- **`n`**, prin care primește un număr natural (**`n`**$\in$ **`[1,10^4]`**);
- **`f`** , prin care furnizează un număr natural cu proprietatea: (**`f-1)·f < n ≤ f·(f+1)`**.

Scrieți definiția completă a subprogramului C/C++.

**Exemplu:** dacă **`n=19`** atunci **`f=4`** (**`3·4 < 19 ≤ 4·5`**). **(10p.)**

## 97. Informatica — Subiectul III, exercițiul 1

**Sursă: 2025 · Speciala C · V06**

*2025_Informatica_Mate-Info_Sesiune_Speciala_C-Cpp_Subiect_V06.pdf*

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

1. Subprogramul **diviz** are un singur parametru, **n**, prin care primește un număr natural (**n**$\in$**[1,10$^9$]**). Subprogramul returnează cel mai mare divizor al lui **n** care este pătrat perfect.
   Scrieți definiția completă a subprogramului C/C++.
   **Exemplu:** pentru **n=72** subprogramul returnează **36**, pentru **n=16** subprogramul returnează **16**, iar pentru **n=15** subprogramul returnează **1**. **(10p.)**

## 98. Informatica — Subiectul III, exercițiul 1

**Sursă: 2025 · Vara Iunie · V01**

*2025_Informatica_Mate-Info_Sesiune_Vara_Iunie-Iulie_C-Cpp_Subiect_V01.pdf*

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

**1.** Numărul natural **an** este **ascendent** al numărului natural **n**, dacă oricare dintre cifrele lui **an** este mai mare sau egală cu cifra unităților lui **n**.

**Exemplu**: oricare dintre numerele **7**, **9**, **98** sau **7998** este ascendent al lui **827**, dar numărul **857** nu este ascendent al lui **827**.

Subprogramul **ascendent** are trei parametri:

- **n**, prin care primește un număr natural ($n \in [0, 10^3)$);
- **x** și **y**, prin care primește câte un număr natural din intervalul $[0, 10^3)$ ($x<y$).

Subprogramul returnează suma ascendenților lui **n** din intervalul **[x,y]**, sau valoarea **0**, dacă nu există niciun astfel de ascendent. Scrieți definiția completă a subprogramului C/C++.

**Exemplu:** dacă **n=827**, **x=9**, **y=800**, subprogramul returnează **7893** (**9+77+78+79+87+88+89+97+98+99+777+778+779+787+788+789+797+798+799=7893**). **(10p.)**

## 99. Informatica — Subiectul III, exercițiul 1

**Sursă: 2026 · Model Oficial**

*2026_Informatica_Mate-Info_Model_Oficial_C-Cpp_Subiect.pdf*

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

1. Subprogramul **Plus** are un singur parametru, **n**, prin care primește un număr natural ($n \in [10, 10^9)$). Subprogramul înlocuiește în **n** fiecare secvență **25** cu câte o secvență **26** și furnizează, prin același parametru, numărul obținut. Dacă nu se înlocuiește nicio secvență, subprogramul furnizează numărul nemodificat.
   Scrieți definiția completă a subprogramului C/C++.
   **Exemplu:** dacă **n=202535250**, după apel **n=202635260**. **(10p.)**

## 100. Informatica — Subiectul III, exercițiul 1

**Sursă: 2026 · Model Oficial**

*2026_Informatica_Mate-Info_Model_Oficial_Pascal_Subiect.pdf*

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

1. Subprogramul **Plus** are un singur parametru, **n**, prin care primește un număr natural ($n \in [10, 10^9)$). Subprogramul înlocuiește în **n** fiecare secvență **25** cu câte o secvență **26** și furnizează, prin același parametru, numărul obținut. Dacă nu se înlocuiește nicio secvență, subprogramul furnizează numărul nemodificat.
   Scrieți definiția completă a subprogramului Pascal.
   **Exemplu:** dacă **n=202535250**, după apel **n=202635260**. **(10p.)**

## 101. Informatica — Subiectul III, exercițiul 1

**Sursă: 2026 · Rezerva Vara Iunie · V05**

*2026_Informatica_Mate-Info_Rezerva_Vara_Iunie-Iulie_C-Cpp_Subiect_V05.pdf*

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

**1.** O editură realizează seturi turistice, formate din câte o broșură de prezentare a zonei locale și câte un catalog al producătorilor locali, astfel încât dacă o broșură are **p** pagini, atunci **p** este un număr prim, iar catalogul din același set are **3·p** pagini.

Subprogramul **pagini** are un parametru, **n**, prin care primește un număr natural (**n**∈**[2,10⁴]**). Subprogramul returnează valoarea **1** dacă, folosind toate cele **n** pagini, se poate realiza un set turistic, sau valoarea **0** în caz contrar. Scrieți în C/C++ definiția completă a subprogramului.

**Exemplu:** dacă **n=44**, subprogramul returnează valoarea **1** (se poate obține o broșură de **11** pagini și un catalog de **33=3·11** pagini), iar dacă **n=48** sau **n=27**, subprogramul returnează **0**. **(10p.)**

## 102. Informatica — Subiectul III, exercițiul 1

**Sursă: 2026 · Speciala C · V04**

*2026_Informatica_Mate-Info_Sesiune_Speciala_C-Cpp_Subiect_V04.pdf*

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

**1.** La o paradă de modă se prezintă modele de rochii și costume, iar ordinea acestora este dată ca o succesiune de cifre, cele impare reprezentând rochii, iar cele pare costume; în prezentare este inclus cel puțin un costum.

Subprogramul **moda** are doi parametri:

- **n**, prin care primește un număr natural din intervalul **[0,10⁹]**, ale cărui cifre, de la stânga la dreapta, corespund modelelor, în ordinea prezentării lor;
- **pc**, prin care furnizează numărul de ordine al primului costum prezentat.

Scrieți definiția completă a subprogramului C/C++.

**Exemplu**: dacă **n=576798** sau **n=5700**, atunci **pc=3**. **(10p.)**

## 103. Informatica — Subiectul III, exercițiul 1

**Sursă: 2026 · Toamna August · V01**

*2026_Informatica_Mate-Info_Sesiune_Toamna_August-Septembrie_C-Cpp_Subiect_V01.pdf*

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

**1.** Un număr este reprezentat în baza **b** dacă cifrele sale aparțin intervalului **[0,b-1]**.
Subprogramul **baza** are doi parametri, **n** și **b**, prin care primește câte un număr natural (**n**∈**[0,10⁹]**, **b**∈**[2,10]**). Subprogramul returnează valoarea **1**, dacă **n** este reprezentat în baza **b**, sau **-1**, în caz contrar.
Scrieți definiția completă a subprogramului C/C++.
**Exemplu:** dacă **n=32412** și **b=8**, subprogramul returnează valoarea **1**, iar dacă **n=32412** și **b=3**, subprogramul returnează valoarea **-1**. **(10p.)**

**2.** Un text, de cel mult **200** de caractere, reprezintă o succesiune de date despre flori de tipuri distincte. Pentru fiecare floare, textul conține câte o pereche formată, în această ordine, din denumire (un șir de cel mult **20** de caractere, litere mici ale alfabetului limbii engleze) și o valoare naturală din intervalul **[1,9]**, reprezentând necesarul zilnic de lumină, dat în număr de ore; fiecare valoare din pereche este urmată de câte un spațiu, ca în exemplu.
Scrieţi un program C/C++ care citeşte de la tastatură un text de tipul precizat și afișează pe ecran denumirea florii cu cel mai mic necesar zilnic de lumină, iar dacă sunt mai multe astfel de flori, se alege prima, în ordine alfabetică.
**Exemplu:** dacă se citește textul **lavanda 7 impatiens 3 iris 5 dalie 6 begonie 3 petunie 7** se afișează pe ecran **begonie** **(10p.)**

**3.** În cadrul testării unor echipamente de comunicații se transmite un șir de biți (**0** sau **1**), care conține cel puțin un bit **1**, iar atât primul cât și ultimul bit sunt **0**. Șirul conține cel puțin un **mesaj de avertisment** de forma unei succesiuni de biți, aflați pe poziții consecutive în șir, dintre care doar unul este **0**. Lungimea mesajului este egală cu numărul biților care îl compun.
Fișierul **bac.txt** conține cel mult **10⁶** numere naturale din mulțimea **{0,1}**, reprezentând șirul de biți transmis. Numerele sunt separate prin câte un spațiu. Se cere să se afișeze pe ecran biții care compun cel mai lung mesaj de avertisment din șirul aflat în fișier. Dacă există mai multe astfel de mesaje, se afișează doar biții unuia dintre acestea. Biții afișați sunt separați prin câte un spațiu. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al spațiului de memorie utilizat.
**Exemplu:** dacă fișierul conține numerele **0 1 1 1 1 1 1 0 0 1 1 1 0 1 1 1 1 0 1 1 0**
se afișează pe ecran **0 1 1 1 1 1 1 1** sau **1 1 1 1 1 1 1 0** sau **1 1 1 0 1 1 1 1**
**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 104. Informatica — Subiectul III, exercițiul 1

**Sursă: 2026 · Toamna August · V01**

*2026_Informatica_Mate-Info_Sesiune_Toamna_August-Septembrie_Pascal_Subiect_V01.pdf*

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

**1.** Un număr este reprezentat în baza **b** dacă cifrele sale aparțin intervalului **[0,b−1]**.
Subprogramul **baza** are doi parametri, **n** și **b**, prin care primește câte un număr natural (**n**∈**[0,10⁹]**, **b**∈**[2,10]**). Subprogramul returnează valoarea **1**, dacă **n** este reprezentat în baza **b**, sau **−1**, în caz contrar.
Scrieți definiția completă a subprogramului Pascal.
**Exemplu:** dacă **n=32412** și **b=8**, subprogramul returnează valoarea **1**, iar dacă **n=32412** și **b=3**, subprogramul returnează valoarea **−1**. **(10p.)**

**2.** Un text, de cel mult **200** de caractere, reprezintă o succesiune de date despre flori de tipuri distincte. Pentru fiecare floare, textul conține câte o pereche formată, în această ordine, din denumire (un șir de cel mult **20** de caractere, litere mici ale alfabetului limbii engleze) și o valoare naturală din intervalul **[1,9]**, reprezentând necesarul zilnic de lumină, dat în număr de ore; fiecare valoare din pereche este urmată de câte un spațiu, ca în exemplu.
Scrieţi un program Pascal care citeşte de la tastatură un text de tipul precizat și afișează pe ecran denumirea florii cu cel mai mic necesar zilnic de lumină, iar dacă sunt mai multe astfel de flori, se alege prima, în ordine alfabetică.
**Exemplu:** dacă se citește textul **lavanda 7 impatiens 3 iris 5 dalie 6 begonie 3 petunie 7** se afișează pe ecran **begonie** **(10p.)**

**3.** În cadrul testării unor echipamente de comunicații se transmite un șir de biți (**0** sau **1**), care conține cel puțin un bit **1**, iar atât primul cât și ultimul bit sunt **0**. Șirul conține cel puțin un **mesaj de avertisment** de forma unei succesiuni de biți, aflați pe poziții consecutive în șir, dintre care doar unul este **0**. Lungimea mesajului este egală cu numărul biților care îl compun.
Fișierul **bac.txt** conține cel mult **10⁶** numere naturale din mulțimea **{0,1}**, reprezentând șirul de biți transmis. Numerele sunt separate prin câte un spațiu. Se cere să se afișeze pe ecran biții care compun cel mai lung mesaj de avertisment din șirul aflat în fișier. Dacă există mai multe astfel de mesaje, se afișează doar biții unuia dintre acestea. Biții afișați sunt separați prin câte un spațiu. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al spațiului de memorie utilizat.
**Exemplu:** dacă fișierul conține numerele **0 1 1 1 1 1 1 0 0 1 1 1 0 1 1 1 1 0 1 1 0**
se afișează pe ecran **0 1 1 1 1 1 1 1** sau **1 1 1 1 1 1 1 0** sau **1 1 1 0 1 1 1 1**
**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul Pascal corespunzător algoritmului proiectat. **(8p.)**

## 105. Informatica — Subiectul III, exercițiul 1

**Sursă: 2026 · Vara Iunie · V03**

*2026_Informatica_Mate-Info_Sesiune_Vara_Iunie-Iulie_C-Cpp_Subiect_V03.pdf*

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

**1.** Numim **număr uniform** asociat unei valori naturale numărul obținut din aceasta prin eliminarea fie a tuturor cifrelor sale pare, fie a tuturor cifrelor sale impare.
**Exemplu:** lui `19472` i se asociază numerele uniforme `197` și `42`.
Subprogramul `ImparPar` are un singur parametru, `n`, prin care primește un număr natural cu toate cifrele nenule ($n \in [11, 10^9)$), având cel puțin o cifră pară și cel puțin o cifră impară. Subprogramul returnează un număr obținut din cifrele numărului uniform impar asociat lui `n`, urmate de cifrele numărului uniform par asociat lui `n`, ca în exemplu. Scrieți definiția completă a subprogramului.
**Exemplu:** dacă `n=19472`, subprogramul returnează numărul `19742`. **(10p.)**

**2.** Scrieți un program C/C++ care citește de la tastatură un cuvânt format din cel puțin două și cel mult `100` de caractere, numai litere mici ale alfabetului englez, și afișează pe ecran mesajul `alterneaza`, dacă în text nu există două vocale sau două consoane pe poziții consecutive, sau mesajul `nu alterneaza`, în caz contrar.
**Exemplu**: pentru cuvintele `imunopolizaharide` sau `pas` se afișează mesajul `alterneaza`
iar pentru cuvântul `aer` se afișează mesajul `nu alterneaza` **(10p.)**

**3.** De-a lungul timpului, o persoană a participat la mai multe concursuri de șah și de go. **Un rezultat motivant** corespunde unei poziții mai bune (cu un număr mai mic) în clasamentul unui concurs, comparativ cu poziția obținută la concursul anterior de același tip.
Fișierul `bac.txt` conține un șir de cel mult $10^4$ numere naturale, reprezentând rezultatele obținute la concursurile la care a participat, în ordine cronologică: pentru fiecare dintre rezultate, câte o pereche formată din poziția ocupată în clasamentul concursului corespunzător (un număr din intervalul `[1,300]`), urmată de valoarea `1`, dacă acesta a fost concurs de șah, sau de valoarea `2`, dacă acesta a fost concurs de go. Fiecare pereche se află pe câte o linie a fișierului, cu valorile separate printr-un spațiu.
Se cere să se afișeze pe ecran, separate printr-un spațiu, două valori naturale, reprezentând, în această ordine, numărul de rezultate motivante obținute la concursurile de șah, respectiv la cele de go. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al spațiului de memorie utilizat.
**Exemplu:** dacă fișierul conține numerele alăturate, pe ecran se afișează `3 2`

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

pentru că la șah a obținut trei rezultate motivante (poziția `3` după poziția `7`, poziția `1` după poziția `5` și poziția `1` după poziția `2`), iar la go două rezultate motivante (poziția `5` după poziția `8`, poziția `1` după poziția `5`).
**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

## 106. Informatica — Subiectul III, exercițiul 1

**Sursă: 2026 · Vara Iunie · V03**

*2026_Informatica_Mate-Info_Sesiune_Vara_Iunie-Iulie_Pascal_Subiect_V03.pdf*

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

**1.** Numim **număr uniform** asociat unei valori naturale numărul obținut din aceasta prin eliminarea fie a tuturor cifrelor sale pare, fie a tuturor cifrelor sale impare.
**Exemplu:** lui **19472** i se asociază numerele uniforme **197** și **42**.
Subprogramul **ImparPar** are un singur parametru, **n**, prin care primește un număr natural cu toate cifrele nenule ($n \in [11, 10^9)$), având cel puțin o cifră pară și cel puțin o cifră impară. Subprogramul returnează un număr obținut din cifrele numărului uniform impar asociat lui **n**, urmate de cifrele numărului uniform par asociat lui **n**, ca în exemplu. Scrieți definiția completă a subprogramului.
**Exemplu:** dacă **n=19472**, subprogramul returnează numărul **19742**. **(10p.)**

**2.** Scrieți un program Pascal care citește de la tastatură un cuvânt format din cel puțin două și cel mult **100** de caractere, numai litere mici ale alfabetului englez, și afișează pe ecran mesajul **alterneaza**, dacă în text nu există două vocale sau două consoane pe poziții consecutive, sau mesajul **nu alterneaza**, în caz contrar.
**Exemplu**: pentru cuvintele **imunopolizaharide** sau **pas** se afișează mesajul **alterneaza**
iar pentru cuvântul **aer** se afișează mesajul **nu alterneaza** **(10p.)**

**3.** De-a lungul timpului, o persoană a participat la mai multe concursuri de șah și de go. **Un rezultat motivant** corespunde unei poziții mai bune (cu un număr mai mic) în clasamentul unui concurs, comparativ cu poziția obținută la concursul anterior de același tip.
Fișierul **bac.txt** conține un șir de cel mult $10^4$ numere naturale, reprezentând rezultatele obținute la concursurile la care a participat, în ordine cronologică: pentru fiecare dintre rezultate, câte o pereche formată din poziția ocupată în clasamentul concursului corespunzător (un număr din intervalul **[1,300]**), urmată de valoarea **1**, dacă acesta a fost concurs de șah, sau de valoarea **2**, dacă acesta a fost concurs de go. Fiecare pereche se află pe câte o linie a fișierului, cu valorile separate printr-un spațiu.
Se cere să se afișeze pe ecran, separate printr-un spațiu, două valori naturale, reprezentând, în această ordine, numărul de rezultate motivante obținute la concursurile de șah, respectiv la cele de go. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al spațiului de memorie utilizat.
**Exemplu:** dacă fișierul conține numerele alăturate, pe ecran se afișează **3 2**

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

pentru că la șah a obținut trei rezultate motivante (poziția **3** după poziția **7**, poziția **1** după poziția **5** și poziția **1** după poziția **2**), iar la go două rezultate motivante (poziția **5** după poziția **8**, poziția **1** după poziția **5**).
**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
**b.** Scrieți programul Pascal corespunzător algoritmului proiectat. **(8p.)**