

## Atestat 2025

### Subiecte propuse pentru proba practică - Programare.

- B) **programare**: presupune rezolvarea de către candidat a unei probleme de programare folosind unul dintre limbajele cunoscute (**C/C++** sau **Pascal**). Candidatul va trebui să prezinte un fișier executabil care rezolvă problema primită.

### Biletul nr. 1

Fișierul `numere.in` conține pe prima linie un număr natural,  $n$  ( $1 < n < 100$ ) și pe a doua linie  $n$  numere naturale, cu cel mult 4 cifre fiecare.

- Să se scrie definiția completă a unui subprogram **Concatenare**, cu doi parametri, numere naturale,  $a$  și  $b$  (de maxim 4 cifre fiecare) care returnează numărul obținut prin concatenarea lui  $a$  cu  $b$ . (ex: dacă subprogramul primește  $a=154$ ,  $b=45$ , va returna **15445**.)
- Scrieți un program care citește datele din fișierul `numere.in` și afișează în fișierul `numere.out`,  $n-1$  numere naturale, folosind apeluri utile ale subprogramului **Concatenare**. Pentru fiecare pereche de numere citite consecutiv, se va afișa cel mai mare număr obținut prin concatenarea celor două.

De exemplu, dacă numerele citite consecutiv sunt **123** și **95**, se va alege cel mai mare dintre cele două numere obținute prin concatenare: dintre 12395 și 95123, se va alege **95123**. Proiectați un algoritm eficient din punctul de vedere al memoriei folosite și al timpului de executare.

**Exemplu:**

<code>numere.in</code>	<code>numere.out</code>
5	312 3128 36128 7836
12 3 128 36 78	

#### Barem de corectare

- subprogramul **Concatenare** **2p**
- declarații corecte, citirea corectă a datelor **1p**
- algoritm de rezolvare corect **4p**
- algoritm eficient **1p**
- afișare corectă a datelor, corectitudinea sintactică a programului **1p**

**Notă:** Se acordă **1 punct** din oficiu.

## Biletul nr. 2

Fișierul `date.in` conține maximum 1000 de numere naturale.

- a. Să se scrie definiția completă a unui subprogram `FactorPutereMax` cu un singur parametru `n`, număr natural. Subprogramul returnează factorul prim cu puterea cea mai mare din descompunerea în factori primi a lui `n`. În cazul în care există factori primi cu puteri egale, atunci se va returna factorul cel mai mare. De exemplu, pentru  $n = 90 = 2^1 * 3^2 * 5^1$ , cea mai mare putere este 2, deci se va returna factorul 3. Pentru  $n = 36 = 2^2 * 3^2$ , puterile fiind egale, se va returna factorul 3.
- b. Scrieți un program C++ care citește cel mult 1000 de numerele naturale din fișierul `date.in` și, folosind apeluri utile ale subprogramului `FactorPutereMax`, scrie în fișierul `date.out`: pe prima linie, pentru fiecare număr citit, factorul prim la puterea cea mai mare, apoi pe linia următoare cel mai mare dintre acești factori.

**Exemplu:**

<code>date.in</code>	<code>date.out</code>
90 24 15 7 36	3 2 5 7 3
	7

### Barem de corectare

- subprogramul `FactorPutereMax` **3p**
- declarații corecte, citirea corectă a datelor **1p**
- algoritm de rezolvare corect **4p**
- afișare corectă a datelor, corectitudinea sintactică a programului **1p**

**Notă:** Se acordă **1 punct** din oficiu.

### Biletul nr. 3

Fișierul `atestat.in` conține maximum 100000 de numere naturale.

- Să se scrie definiția completă a unui subprogram `CifraMax`, cu un singur parametru `n`, număr natural. Subprogramul returnează cea mai mare cifră a numărului `n`. De exemplu, pentru `n = 65706`, se va returna 7.
- Scrieți un program C++ care citește numerele din `atestat.in` și, folosind apeluri utile ale subprogramului `CifraMax`, scrie în fișierul `atestat.out` cifra care este maximă pentru cele mai multe numere din fișier. Dacă există mai multe astfel de cifre, se vor afișa toate cu această proprietate, în ordine crescătoare. Proiectați un algoritm eficient din punctul de vedere al memoriei folosite și al timpului de executare.

**Exemplu:**

<code>atestat.in</code>	<code>atestat.out</code>	Explicație
458 188 956 659 88	8	8 este cifra maximă pentru 3 din numere, 9 este cifra maximă pentru doua numere
125 54 125 65 126 62	5 6	5 este cifra maximă pentru 3 din numere, 6 este cifra maximă tot pentru 3 numere

#### Barem de corectare

- subprogramul `CifraMax` **1p**
- declarații corecte, citirea corectă a datelor **1p**
- algoritm de rezolvare corect **5p**
- algoritm eficient **1p**
- afișare corectă a datelor, corectitudinea sintactică a programului **1p**

**Notă:** Se acordă **1 punct** din oficiu.

### Biletul nr. 4

- a. Să se scrie definiția completă a unui subprogram **Inversez** cu trei paramentrii: **v** – tablou unidimensional cu maximum 100 de elemente, **numere reale**, si doi parametrii întregi: **p**, **q** - două poziții din tabloul **v**,  $p < q < 100$  . Subprogramul modifică tabloul prin inversarea elementelor între poziția **p** și poziția **q** și returnează prin același parametru tabloul modificat. (Dacă  $v = \{1.2, 2, 3.1, 4.25, 5, 6.3\}$ ,  $p = 2$ ,  $q = 5$ , după inversare tabloul va avea conținutul:  $v = \{1.2, 5, 4.25, 3.1, 2, 6.3\}$  .
- b. Să se scrie definiția completă a unui subprogram **Stergp** cu trei paramentrii: **v** – tablou unidimensional cu maximum 100 de elemente, **numere reale**, si doi parametrii întregi: **n**, **p** – numărul de elemente din vector și  $p < 100$  . Subprogramul modifică tabloul prin ștergerea elementului de pe poziția **p** și returnează prin același parametru tabloul modificat.
- c. Scrieți un program C++ care citește din fișierul **numere.in** de pe prima linie număr natural, **n** ( $n < 100$ ) și de pe a doua linie un șir cu maxim 100 de numere **reale**. Folosind apeluri utile ale subprogramului **Stergp**, șterge elementul din mijlocul șirului, în cazul în care acesta are un număr impar de elemente. Folosind apeluri utile ale subprogramului **Inversez**, se vor inversa fiecare din cele două jumătăți ale șirului, apoi afișează, în fișierul **numere.out**, conținutul șirului modificat.

#### Exemplu:

numere.in	numere.out
7	4.3 8.2 4.1 7.9 3.1 2.1
4.1 8.2 4.3 <u>8.1</u> 2.1 3.1 7.9	

#### Barem de corectare

- subprogramul **Inversez** 2p
- subprogramul **Stergp** 2p
- declarații corecte, citirea corectă a datelor 1p
- algoritm de rezolvare corect 2p
- algoritm eficient 1p
- afișare corectă a datelor, corectitudinea sintactică a programului 1p

**Notă:** Se acordă 1 punct din oficiu.

**Biletul nr. 5**

- a. Să se scrie definiția completă a unui subprogram **CifrePare** cu un paramentru **n**, număr natural, care returnează numărul format doar din cifrele pare ale lui **n**, având cifrele în ordinea inițială. (Ex: pentru  $n = 12245$ , se returnează 224)
- b. Să se scrie definiția completă a unui subprogram **CifreCresc** cu un paramentru **n**, număr natural, care returnează 1 dacă cifrele lui **n** sunt în ordine crescătoare și 0 altfel. (Ex: pentru  $n = 12245$ , se returnează 1, iar pentru 12436, returnează 0)
- c. Fișierul **date.in** conține maximum 1000000 de numere naturale, fiecare din ele conținând cel puțin o cifră pară. Scrieți un program C++ care citește numerele din **date.in** și, folosind apeluri utile ale celor două subprograme, scrie în fișierul **date.out** suma formată prin adunarea numerelor obținute din cifrele pare ale numerelor citite, dacă numerele astfel obtinute au cifrele în ordine crescătoare.

<b>date.in</b>	<b>date.out</b>	<b>Explicație</b>
1546    65423    26    623 2318	100	46+26+28=100

**Barem de corectare**

- subprogramul CifrePare 2p
- subprogramul CifreCresc 2p
- declarații corecte, citirea corectă a datelor 1p
- algoritm de rezolvare corect 3p
- afișare corectă a datelor, corectitudinea sintactică a programului 1p

**Notă:** Se acordă **1 punct** din oficiu.

**Biletul nr. 6**

- a. Să se scrie definiția completă a unui subprogram **Cmmdc** cu doi parametri, numere naturale  $a$  și  $b$ , care returnează cel mai mare divizor comun al numerelor  $a$  și  $b$  (Pentru 20 și 8, returnează 4)
- b. Să se scrie definiția completă a unui subprogram **Cmmm** cu doi parametri, numere naturale  $a$  și  $b$ , care returnează cel mai mic multiplu comun al numerelor  $a$  și  $b$  (Pentru 20 și 8, returnează 40)
- c. Fișierul **atestat.in** conține maximum 1000000 de numere naturale. Scrieți un program C++ care citește numerele din **atestat.in** și, folosind apeluri utile al subprogramelor **Cmmdc** și **Cmmm**, scrie pe prima linie a fișierului **atestat.out** cel mai mare divizor comun al numerelor citite din fișier, iar pe a doua, cel mai mic multiplu comun al numerelor citite din fișier.

atestat.in	atestat.out
16 32 8 4 18	2
	288

**Barem de corectare**

- subprogramul **Cmmdc** 1.5p
- subprogramul **Cmmm** 1.5p
- declarații corecte, citirea corectă a datelor 1p
- algoritm de rezolvare corect 4p
- afișare corectă a datelor, corectitudinea sintactică a programului 1p

**Notă:** Se acordă **1 punct** din oficiu.

### Biletul nr. 7

- a. Să se scrie definiția completă a unui subprogram **InsereszP** cu patru parametrii: **n** – numărul de elemente al vectorului **v**, **v** – **vector** cu maximum 100 de elemente, numere întregi, **p**, **p**  $\leq 100$  reprezentând o poziție în vector și **val** – o valoare întreagă. Subprogramul modifică vectorul prin inserarea, pe poziția **p** a unui element de valoare **val** și returnează prin parametrul **v** vectorul modificat. (Ex: pentru  $n=4$ ,  $v=\{2, 5, 6, 9\}$ ,  $p = 2$ ,  $val = 100$ , vectorul modificat este:  $\{2, 100, 5, 6, 9\}$ , considerând că indexarea s-a făcut de la 1)
- b. Să se scrie definiția completă a unui subprogram **SumaCifre**, cu un singur parametru **x**, număr natural. Subprogramul returnează suma cifrelor lui **x**. De exemplu, pentru  $n = 27$ , se va returna 9.
- c. Fișierul **numere.in** conține pe prima linie **n**, număr natural,  $n < 100$ , și pe a doua linie maximum 100 de numere naturale. Scrieți un program C++ care citește datele din **numere.in** și, folosind apeluri utile ale subprogramelor **InsereszP** și **SumaCifre**, inserează în vector, după fiecare element, suma cifrelor sale, apoi afișează în **numere.out** vectorul modificat.

<b>numere.in</b>	<b>numere.out</b>
5	48 12 13 4 6 6 91 10 30 3
48 13 6 91 30	

#### Barem de corectare

- subprogramul Inversez 2p
- subprogramul SumaCifre 1p
- declarații corecte, citirea corectă a datelor 1p
- algoritm de rezolvare corect 4p
- afișare corectă a datelor, corectitudinea sintactică a programului 1p

**Notă:** Se acordă **1 punct** din oficiu.

### Biletul nr. 8

- a. Să se scrie definiția completă a unui subprogram **EPrim**, cu un singur parametru **x**, număr natural, care returnează 1 dacă numărul este prim și 0 în caz contrar. De exemplu, pentru  $n = 27$ , se va returna 0, iar pentru 13 se returnează 1.
- b. Fișierul **numere.in** conține cel mult 100000 de numere naturale, fiecare având cel mult două cifre. Scrieți un program C++ care citește datele din **numere.in** și, utilizând apeluri ale subprogramului **EPrim**, determina numărul prim din fișier care apare de cele mai multe ori în fișier. Programul afișează în fișierul **numere.out**, numărul prim determinat și numărul său de apariții, sau mesajul „Nu exista”, dacă în fișier nu se găsește niciun număr prim.

<b>numere.in</b>	<b>numere.out</b>
48 11 5 13 6 11 11 30	11 3
44 98 22 14 852	Nu exista

#### Barem de corectare

- subprogramul Eprim 2p
- declarații corecte, citirea corectă a datelor 1p
- algoritm de rezolvare corect 5p
- afișare corectă a datelor, corectitudinea sintactică a programului 1p

**Notă:** Se acordă **1 punct** din oficiu.

### Biletul nr. 9

- a. Să se scrie definiția completă a unui subprogram **CifreDistincte** cu un paramentru **n**, număr natural, care returnează **1** dacă numărul **n** are toate cifrele distincte, și **0** în caz contrar. (Ex: pentru  $n = 12245$ , se returnează 0, iar pentru  $n = 2659$  se returnează 1.)
- b. Fișierul **date.in** conține maximum 1000 de numere naturale. Scrieți un program C++ care citește toate numerele din **date.in** și, folosind apeluri utile ale subprogramului **CifreDistincte**, scrie în fișierul **date.out** pe prima linie, toate numerele citite care au cifre distincte, urmate de numărul lor (câte au fost găsite), iar pe a doua linie a fișierului numerele citite care nu au cifre distincte, urmate de numărul lor. Dacă una din cele două categorii de numere nu există, se va afișa un mesajul "Nu exista" pe linia corespunzătoare.

<b>date.in</b>	<b>date.out</b>
<b>1546 65423 6526 623</b> 2311 3263	<b>1546 65423 623 3</b> 6526 2311 3263 <b>3</b>
12345 6598 458	12345 6598 458 3 Nu există
25454 6988 7877	Nu există 25454 6988 7877 3

#### Barem de corectare

- subprogramul CifreDistincte **3p**
- declarații corecte, citirea corectă a datelor **1p**
- algoritm de rezolvare corect **4p**
- afișare corectă a datelor, corectitudinea sintactică a programului **1p**

**Notă:** Se acordă **1 punct** din oficiu.

### Biletul nr. 10

- a. Să se scrie definiția completă a unui subprogram **PozitiiPare**, cu 4 parametri: **n** – numărul de elemente ale unui tablou unidimensional, **v** – tablou unidimensional cu elemente numere întregi, **prim**, **ultim** – doi parametri de ieșire prin care subprogramul furnizează pozițiile primului respectiv al ultimului element par din tabloul **v**. (Ex  $v = \{1, 3, 4, 3, 8, 10, 6, 5, 3\}$ , se va calcula  $prim = 3, ultim = 7$ )
- b. Scrieți un subprogram **Maxim** care, primind ca parametrii: **n**, numărul de elemente ale unui vector, **v** - vectorul si doua poziții întregi, **p** si **q**, determină elementul maxim din vector între pozițiile **p** si **q**. ( $p < q < n$ )
- c. Fișierul **numere.in** conține pe prima linie **n**, număr natural,  $n < 100$ , și pe a doua linie maximum 100 de numere întregi, dintre care cel puțin două sunt pare. Scrieți un program C++ care citește din fișierul **numere.in** un număr natural **n** ( $n < 100$ ), apoi un șir cu **n** numere întregi având cel mult 9 cifre fiecare, și utilizând apeluri ale celor doua subprograme, determină elementul maxim aflat între primul și ultimul număr par. Programul afișează în **numere.out** elementul cerut.

numere.in	numere.out
7	
47 <u>52</u> 13    6    91 <u>10</u> 301	<b>91</b>

#### Barem de corectare

- subprogramul PozitiiPare 3p
- subprogramul Sortpq 2p
- declarații corecte, citirea corectă a datelor 1p
- algoritm de rezolvare corect 2p
- afișare corectă a datelor, corectitudinea sintactică a programului 1p

**Notă:** Se acordă **1 punct** din oficiu.

**Biletul nr. 11**

- c. Să se scrie definiția completă a unui subprogram **Nrdivizori**, cu un singur parametru **x**, număr natural, care returnează numărul de divizori naturali ai lui **x**.
- d. Fișierul **numere.in** conține cel mult 100000 de numere naturale, fiecare având cel mult două cifre. Scrieți un program C++ care citește datele din **numere.in** și, utilizând apeluri ale subprogramului **Nrdivizori**, determina numărul din fișier cu cei mai mulți divizori naturali. Programul afișează în fișierul **numere.out**, numărul determinat.

<b>numere.in</b>	<b>numere.out</b>
48 11 5 13 6 11 11 30	48

**Barem de corectare**

- subprogramul Eprim 2p
- declarații corecte, citirea corectă a datelor 1p
- algoritm de rezolvare corect 5p
- afișare corectă a datelor, corectitudinea sintactică a programului 1p

**Notă:** Se acordă **1 punct** din oficiu.

**Subiecte propuse pentru obținerea atestatului profesional în informatică - 2025**  
**Programare**

---

Subiectele au fost propuse și aprobate de către Consiliul Consultativ al profesorilor de informatică din județul Bistrița-Năsăud