

## PROGRAMARE SPECIALIZAREA MATEMATICĂ – INFORMATICĂ, NEINTENSIV INFORMATICĂ

### Subiectul nr. 1

Fișierul `atestat.in` conține două linii. Pe prima linie este scris un număr natural nenul  $n$ , ( $5 < n < 30$ ). Pe cea de-a doua linie a fișierului sunt scrise  $n$  numere naturale separate prin câte un spațiu, formate fiecare din cel mult 4 cifre, reprezentând un șir de  $n$  numere naturale nenule. Cel puțin unul din cele  $n$  numere din șir este număr impar.

Să se scrie un program, în limbajul Pascal/C/C++, care:

- a) afișează pe ecran, în linie, separate printr-un spațiu, toate numerele impare din șir;
- b) scrie în fișierul `atestat.out` cel mai mare divizor comun al elementului minim și cel maxim din șir.

#### **Exemplu:**

<code>atestat.in</code>	Date de ieșire:
6 1232 543 98 233 242 33	a) 543 233 33 b) Fișierul <code>atestat.out</code> conține: 11

**Subiectul nr. 2**

Fișierul `atestat.in` conține două linii. Pe prima linie este scris un număr natural nenul  $n$ , ( $5 < n < 30$ ). Pe cea de-a doua linie a fișierului sunt scrise  $n$  numere naturale separate prin câte un spațiu, formate fiecare din cel mult 4 cifre, reprezentând un șir de  $n$  numere naturale. Cel puțin unul din cele  $n$  numere din șir are cifra unităților mai mare decât numărul 5.

Să se scrie un program, în limbajul Pascal/C/C++, care:

- afișează pe ecran, în linie, separate printr-un spațiu, toate numerele care au ultima cifră mai mare decât numărul 5;
- scrie în fișierul `atestat.out`, separate prin câte un spațiu, toate numerele *prime* din șirul dat. Un număr natural este *prim* dacă are exact doi divizori distincți (numărul 1 și numărul în sine). Dacă nu există numere *prime* se va scrie pe prima linie a fișierului mesajul **“NU EXISTA NUMERE PRIME”**.

**Exemplu:**

<code>atestat.in</code>	Date de ieșire:
6 128 2715 17 1009 61 2715	a) 128 17 1009 b) Fișierul <code>atestat.out</code> conține: 17 1009 61

**Subiectul nr. 3**

Fișierul `atestat.in` conține două linii. Pe prima linie este scris un număr natural nenul  $n$ , ( $5 < n < 30$ ). Pe cea de-a doua linie a fișierului sunt scrise  $n$  numere naturale separate prin câte un spațiu, formate fiecare din cel mult 4 și cel puțin 2 cifre, reprezentând un șir de  $n$  numere naturale. Cel puțin unul din cele  $n$  numere este format din 2 cifre.

Să se scrie un program, în limbajul Pascal/C/C++, care:

- afișează pe ecran, în linie, separate printr-un spațiu, toate numerele formate din două cifre;
- scrie în fișierul `atestat.out` pe linii diferite, numerele din șir care au proprietatea ca printre numerele precedente să existe cel puțin un număr strict mai mic decât valoarea lor. Dacă nu există astfel de numere se va scrie pe prima linie a fișierului mesajul **“NU EXISTA ASTFEL DE NUMERE”**.

**Exemplu:**

<code>atestat.in</code>	Date de ieșire:
6	a) 35 32 46
364 35 32 472 46 1222	b) Fișierul <code>atestat.out</code> conține:
Se citesc de la tastatură	472
a=100	46
b=1000	1222

**Subiectul nr. 4**

Fișierul `atestat.in` conține două linii. Pe prima linie este scris un număr natural nenul  $n$ , ( $5 < n < 30$ ). Pe cea de-a doua linie a fișierului sunt scrise  $n$  numere naturale separate prin câte un spațiu, formate fiecare din cel mult 4 cifre și cel puțin 2 cifre, reprezentând un șir de  $n$  numere naturale. Cel puțin unul din cele  $n$  numere din șir este număr par.

Să se scrie un program, în limbajul Pascal/C/C++, care:

- afișează pe ecran, în linie, separate printr-un spațiu, toate numerele pare din șir;
- scrie în fișierul `atestat.out`, pe o linie separate prin câte un spațiu, toate numerele *perfecte* din șir. Un număr natural se numește *perfect* dacă suma divizorilor săi, exceptând numărul însuși, este egală cu numărul dat, de exemplu:  $28 = 1 + 2 + 4 + 7 + 14$ . Dacă nu există astfel de numere, se va scrie pe prima linie a fișierului mesajul "NU EXISTA NUMERE PERFECTE".

**Exemplu:**

<code>atestat.in</code>	Date de ieșire:
6 28 15 496 5568 345 1289	a) 28 496 5568 b) Fișierul <code>atestat.out</code> conține: 496 28

**Subiectul nr. 5**

Fișierul `atestat.in` conține trei linii. Pe prima linie este scris un număr natural nenul  $n$ , ( $5 < n < 30$ ). Pe cea de-a doua linie a fișierului sunt scrise  $n$  numere naturale, separate prin câte un spațiu, formate fiecare din cel mult 4 cifre și cel puțin 2 cifre, reprezentând un șir de  $n$  numere naturale.

Primul număr din șir se găsește pe poziția 1, iar ultimul număr din șir se găsește pe poziția  $n$ . Cel puțin unul din cele  $n$  numere din șir are cifra zecilor impară. Pe ultima linie este scris un număr natural  $k$ ,  $1 \leq k < n$ .

Să se scrie un program, în limbajul Pascal/C/C++, care:

- afișează pe ecran, în linie, separate printr-un spațiu, toate numerele din șir, care au cifra zecilor impară;
- scrie în fișierul `atestat.out` pe o linie, separate prin câte un spațiu, toate numerele *supraperfecte* din șirul rezultat la punctul b. Un număr natural se numește *supraperfect* dacă suma divizorilor săi, exceptând numărul însuși, este mai mare decât numărul dat, de exemplu:  $18 < 1+2+3+6+9$ . Dacă nu există astfel de numere, se va scrie pe prima linie a fișierului mesajul **"NU EXISTA NUMERE SUPRAPERFECTE"**.

**Exemplu:**

<code>atestat.in</code>	Date de ieșire:
6	a) 6138 12 496 18
6138 12 496 2523 18 1243	b) Fișierul <code>atestat.out</code> conține:
3	6138 12 18

**Subiectul nr. 6**

Fișierul `atestat.in` conține două linii. Pe prima linie este scris un număr natural nenul  $n$ , ( $5 < n < 20$ ). Pe cea de-a doua linie a fișierului sunt scrise  $n$  numere reale separate prin câte un spațiu, reprezentând un șir de  $n$  numere reale. Cel puțin unul din cele  $n$  numere din șir este număr strict pozitiv.

Să se scrie un program în limbajul Pascal/C/C++, care:

- afișează pe ecran, în linie, separate printr-un spațiu, cu exact două zecimale toate numerele strict pozitive din șir;
- scrie în fișierul `atestat.out` media aritmetică a numerelor negative din șir. Media aritmetică se va afișa cu o precizie de două zecimale. Dacă nu există numere negative, se va scrie pe prima linie a fișierului mesajul "NU EXISTA NUMERE NEGATIVE".

**Exemplu:**

<code>atestat.in</code>
6
13.34567 -24.7491 123.932 5.5673 -41.2349 -562

Date de ieșire:
a) 13.35 123.93 5.57
b) Fișierul <code>atestat.out</code> conține: -209.33

**Subiectul nr. 7**

Fișierul `atestat.in` conține trei linii. Pe prima linie este scris un număr natural nenul  $n$ , ( $5 < n < 30$ ). Pe cea de-a doua linie a fișierului sunt scrise  $n$  numere naturale separate prin câte un spațiu, formate fiecare din cel mult 4 cifre și cel puțin 2 cifre, reprezentând un șir de  $n$  numere naturale. Primul număr din șir se găsește pe poziția 1, iar ultimul număr din șir se găsește pe poziția  $n$ . Cel puțin unul din cele  $n$  numere din șir este divizibil cu numărul 7. Pe ultima linie este scris un număr natural  $k$ ,  $1 \leq k \leq n$ .

Să se scrie un program în limbajul Pascal/C/C++, care:

- afișează pe ecran, în linie, separate printr-un spațiu, numerele din șir divizibile cu numărul 7;
- scrie în fișierul `atestat.out`, numărul obținut prin însumarea ultimei cifre a fiecărui număr din șirul dat.

**Exemplu:**

<code>atestat.in</code>	Date de ieșire:
6	a) 182 546
20 182 1027 546 1007 36	b) Fișierul <code>atestat.out</code> conține:
3	28

**Subiectul nr. 8**

Fișierul `atestat.in` conține trei linii. Pe prima linie este scris un număr natural nenul  $n$ , ( $5 < n < 30$ ). Pe cea de-a doua linie a fișierului sunt scrise  $n$  numere naturale separate prin câte un spațiu, formate fiecare din cel mult 4 cifre și cel puțin 2 cifre, reprezentând un șir de  $n$  numere naturale. Cel puțin unul din cele  $n$  numere din șir este număr impar. Pe ultima linie este scris un număr natural  $k$ , format din cel mult 4 cifre și cel puțin 2 cifre.

Să se scrie un program în limbajul Pascal/C/C++, care:

- afișează pe ecran, în linie, separate printr-un spațiu, numerele impare din șir;
- pe scrie în fișierul, `atestat.out`, în linie, separate prin câte un spațiu, *cifrele de control* ale elementelor șirului dat. *Cifra de control* a unui număr natural se obține prin însumarea cifrelor din scrierea zecimală a numărului, apoi se însumează cifrele acestei sume și tot așa până se obține o sumă formată dintr-o singură cifră. Exemplu: pentru numărul 672 avem  $6+7+2=15$ ,  $1+5=6$ , cifra de control este 6.

**Exemplu:**

<code>atestat.in</code>	Date de ieșire:
6	a) 123 1245 8345
123 1245 34 672 8345 34	b) Fișierul <code>atestat.out</code>
34	conține:
	6 3 7 6 2 7



**Subiectul nr. 9**

Fișierul `atestat.in` conține două linii. Pe prima linie este scris un număr natural nenul  $n$ , ( $5 < n < 30$ ). Pe cea de-a doua linie a fișierului sunt scrise  $n$  numere naturale separate prin câte un spațiu, formate fiecare din cel mult 4 cifre și cel puțin 2 cifre, reprezentând un șir de  $n$  numere naturale. Cel puțin unul din cele  $n$  numere din șir este divizibil cu numărul 11.

Să se scrie un program în limbajul Pascal/C/C++, care:

- afișează pe ecran, în linie, separate printr-un spațiu, toate numerele din șir divizibile cu numărul 11;
- scrie în fișierul `atestat.out`, separate prin câte un spațiu, elementul maxim și poziția acestuia în șir. Dacă există mai multe numere egale cu maximum atunci se va afișa poziția ultimului.

**Exemplu:**

<code>atestat.in</code>	Date de ieșire:
6 123 8445 33 672 8445 144	a) 33 b) Fișierul <code>atestat.out</code> conține: 8445 5

**Subiectul nr. 10**

Fișierul `atestat.in` conține două linii. Pe prima linie este scris un număr natural nenul  $n$ , ( $5 < n < 20$ ). Pe cea de-a doua linie a fișierului sunt scrise  $n$  numere reale separate prin câte un spațiu, reprezentând un șir de  $n$  numere reale. Cel puțin unul din cele  $n$  numere din șir este strict negativ.

Să se scrie un program în limbajul Pascal/C/C++, care:

- afișează pe ecran, în linie, separate printr-un spațiu, cu exact două zecimale, toate numerele strict negative din șir,
- scrie în fișierul `atestat.out`, pe prima linie, separate prin câte un spațiu, elementul minim cu exact două zecimale și poziția acestuia în șirul ordonat obținut la punctul b). Dacă există mai multe numere egale cu minimul atunci se va afișa poziția primului.

**Exemplu:**

<code>atestat.in</code>
6
13.34567 -721.2349 -24.7491 123.932 5.5673 -721.2349

Date de ieșire:
a) -721.23 -24.75 -721.23
b) Fișierul <code>atestat.out</code> conține: -721.23 5

**Subiectul nr. 11**

Fișierul `atestat.in` conține pe prima linie numerele naturale nenule  $m$  și  $n$  ( $2 < m, n < 20$ ), care reprezintă numărul de linii, respectiv numărul de coloane ale unei matrice  $A$ . Pe fiecare din următoarele  $m$  linii, sunt scrise câte  $n$  numere naturale, formate din cel mult 4 cifre, separate prin câte un spațiu, reprezentând valorile elementelor matricei  $A$ .

Să se scrie un program, în limbajul Pascal/C/C++, care:

- afișează pe ecran elementele matricei  $A$ , linie cu linie, elementele de pe aceeași linie fiind separate prin câte un spațiu;
- scrie în fișierul `atestat.out`, mesajul „DA” în cazul în care coloana cu indicele  $k$  are toate elementele nule sau mesajul „NU” în caz contrar. Numărul natural  $k$  se citește de la tastatură  $k \in [\text{indice prima coloană}, \text{indice ultima coloană}]$ .

**Exemplu:**

<code>atestat.in</code>	Date de ieșire:
<code>3 4</code> <code>124 0 42 18</code> <code>333 0 1096 220</code> <code>41 0 97 15</code> <code>Se citește de la tastatură</code> <code>k=3</code>	a) <code>124 0 42 18</code> <code>333 0 1096 220</code> b) Fișierul <code>atestat.out</code> conține: <code>NU</code>

**Subiectul nr. 12**

Fișierul `atestat.in` conține pe prima linie numerele naturale nenule  $m$  și  $n$  ( $2 < m, n < 20$ ), care reprezintă numărul de linii, respectiv numărul de coloane ale unei matrice  $A$ . Pe fiecare din următoarele  $m$  linii, sunt scrise câte  $n$  numere naturale, formate fiecare din cel mult 4 cifre și cel puțin 2 cifre, separate prin câte un spațiu, reprezentând valorile elementelor matricei  $A$ .

Să se scrie un program, în limbajul Pascal/C/C++, care:

- afișează pe ecran elementele matricei  $A$ , linie cu linie, elementele de pe aceeași linie fiind separate prin câte un spațiu;
- scrie în fișierul `atestat.out`, pe prima linie, separate prin câte un spațiu, toate numerele *palindrom* din matricea  $A$ . Un număr natural, format din minimum două cifre, este *palindrom* dacă numărul citit de la stânga la dreapta este egal cu numărul citit de la dreapta la stânga, de exemplu: 11, 121, 2112. Dacă nu există astfel de numere se va scrie pe prima linie a fișierului mesajul "NU EXISTA NUMERE PALINDROM".

**Exemplu:**

<code>atestat.in</code>	Date de ieșire:
4 3 15 36 1221 557 291 7603 953 34 232 678 43 400	a) 15 36 1221 557 291 7603 953 34 232 678 43 400 b) Fișierul <code>atestat.out</code> conține: 1221 232

**Subiectul nr. 13**

Fișierul `atestat.in` conține pe prima linie numerele naturale nenule  $m$  și  $n$  ( $2 < m, n < 20$ ), care reprezintă numărul de linii, respectiv numărul de coloane ale unei matrice  $A$ . Pe fiecare din următoarele  $m$  linii, sunt scrise câte  $n$  numere întregi, formate din cel mult 4 cifre, separate prin câte un spațiu, reprezentând valorile elementelor matricei  $A$ .

Să se scrie un program, în limbajul Pascal/C/C++, care:

- afișează pe ecran elementele matricei  $A$ , linie cu linie, elementele de pe aceeași linie fiind separate prin câte un spațiu;
- scrie în fișierul `atestat.out`, mesajul „DA” dacă suma elementelor de pe coloana cu indicele  $k$  este un număr impar sau mesajul „NU” în caz contrar. Numărul natural  $k$  se citește de la tastatură,  $k \in [\text{indice prima coloană}, \text{indice ultima coloană}]$ .

**Exemplu:**

<code>atestat.in</code>	Date de ieșire:
<code>4 3</code>	a) <code>111 34 -42</code>
<code>111 34 -42</code>	<code>569 -92 -23</code>
<code>569 -92 -23</code>	<code>-3 744 2345</code>
<code>-3 744 2345</code>	<code>89 -8 942</code>
<code>89 -8 942</code>	b) Fișierul <code>atestat.out</code> conține:
<code>Se citește de la tastatură</code>	<code>NU</code>
<code>k=2</code>	

**Subiectul nr. 14**

Fișierul `atestat.in` conține pe prima linie numerele naturale nenule  $m$  și  $n$  ( $2 < m, n < 20$ ), care reprezintă numărul de linii, respectiv numărul de coloane ale unei matrice  $A$ . Pe fiecare din următoarele  $m$  linii, sunt scrise câte  $n$  numere întregi, formate din cel mult 4 cifre, separate prin câte un spațiu, reprezentând valorile elementelor matricei  $A$ .

Să se scrie un program, în limbajul Pascal/C/C++, care:

- afișează pe ecran, în linie, elementele ultimei coloane a matricei  $A$ , separate prin câte un spațiu;
- scrie în fișierul `atestat.out`, numărul obținut prin adunarea ultimei cifre a fiecărui număr din matricea  $A$ .

**Exemplu:**

<code>atestat.in</code>	Date de ieșire:
3 4 -9 234 2 -37 567 -88 123 234 34 9 -5 67	a) -37 234 67 b) Fișierul <code>atestat.out</code> conține: 69

**Subiectul nr. 15**

Fișierul `atestat.in` conține pe prima linie numerele naturale nenule  $m$  și  $n$  ( $2 < m, n < 20$ ), care reprezintă numărul de linii, respectiv numărul de coloane ale unei matrice  $A$ . Pe fiecare din următoarele  $m$  linii, sunt scrise câte  $n$  numere reale pozitive, formate din cel mult 6 cifre, separate prin câte un spațiu, reprezentând valorile elementelor matricei  $A$ .

Să se scrie un program, în limbajul Pascal/C/C++, care:

- afișează pe ecran elementele matricei  $A$ , linie cu linie, elementele de pe aceeași linie fiind separate prin câte un spațiu;
- scrie în fișierul `atestat.out`, pe prima linie, suma dintre partea întreagă a elementului maxim și partea întreagă a elementului minim din matricea  $A$ .

**Exemplu:**

<code>atestat.in</code>	Date de ieșire:
4 3	a) 678.92 12.34 44.5678
678.92 12.34 44.5678	9.1 5.5 32.1
9.1 5.5 32.1	34.997 11.2 79.22
34.997 11.2 79.22	0.2 94.2 23.32
0.2 94.2 23.32	b) Fișierul <code>atestat.out</code> conține: 678

**Subiectul nr. 16**

Fișierul `atestat.in` conține pe prima linie numărul  $n$ , natural nenul ( $2 < n < 20$ ), ce reprezintă numărul de linii ale unei matrice pătratice  $A$ . Pe fiecare din următoarele  $n$  linii, sunt scrise câte  $n$  numere naturale, formate din cel mult 4 cifre, separate prin câte un spațiu, reprezentând valorile elementelor matricei  $A$ .

Să se scrie un program, în limbajul Pascal/C/C++, care:

- afișează pe ecran, în linie, suma elementelor diagonalei principale a matricei  $A$ ;
- scrie în fișierul `atestat.out`, pe prima linie, separate prin câte un spațiu, toate numerele *pătrate perfecte* din matricea  $A$ . Un număr natural  $a$  se numește *pătrat perfect* dacă există un alt număr întreg  $b$  astfel încât  $a=b^2$ , de exemplu  $25=5^2$ . Dacă nu există numere pătrate perfecte se va scrie pe prima linie a fișierului mesajul "NU EXISTA NUMERE PATRATE PERFECTE".

**Exemplu:**

<code>atestat.in</code>	Date de ieșire:
4 654 345 21 1024 7 25 567 33 12 765 2 89 43 678 812 7921	a) 8602 b) Fișierul <code>atestat.out</code> conține: 1024 25 7921



**Subiectul nr. 17**

Fișierul `atestat.in` conține pe prima linie numărul  $n$ , natural nenul ( $2 < n < 20$ ), ce reprezintă numărul de linii ale unei matrice pătratice  $A$ . Pe fiecare din următoarele  $n$  linii, sunt scrise câte  $n$  numere naturale, formate din cel mult 4 cifre, separate prin câte un spațiu, reprezentând valorile elementelor matricei  $A$ .

Să se scrie un program, în limbajul Pascal/C/C++, care:

- afișează pe ecran, în linie, elementele diagonalei secundare a matricei  $A$ , separate prin câte un spațiu;
- scrie în fișierul `atestat.out`, pe prima linie, separate prin câte un spațiu, toate numerele *cuburi perfecte* din matricea  $A$ . Un număr întreg  $a$  se numește cub perfect dacă există un alt număr întreg  $b$  astfel încât  $a=b^3$ , de exemplu  $125=5^3$ . Dacă nu există numere cuburi perfecte se va scrie pe prima linie a fișierului mesajul "NU EXISTA NUMERE CUBURI PERFECTE".

**Exemplu:**

<code>atestat.in</code>	Date de ieșire:
4 456 345 5832 1024 343 25 567 33 24 765 12 121 43 678 812 1728	a) 1024 567 765 43 b) Fișierul <code>atestat.out</code> conține: 5832 343 1728

**Subiectul nr. 18**

Fișierul `atestat.in` conține o singură linie pe care se află scris un text format din cel mult 70 de caractere, primul caracter fiind literă, iar cuvintele din text sunt separate prin câte un spațiu. Fiecare cuvânt este format doar din litere mari sau mici ale alfabetului limbii engleze.

Să se scrie un program în limbajul Pascal/C/C++, care:

- afișează textul pe o singură linie a ecranului, transformat astfel: litere mari în litere mici și litere mici în litere mari;
- scrie în fișierul `atestat.out`, pe un rând, separate printr-un spațiu, toate cuvintele din fișierul `atestat.in`, care încep cu literă mare. Dacă nu există cuvinte care să înceapă cu literă mare se va scrie pe prima linie a fișierului mesajul „**NU EXISTA ASTFEL DE CUVINTE**”.

**Exemplu:**

<code>atestat.in</code>
Vacanta de VARA o voi Petrece la MUNTE

Date de ieșire:
a) vACANTA DE vara O VOI pETRECE LA munte
b) Fișierul <code>atestat.out</code> conține: Vacanta VARA Petrece MUNTE

**Subiectul nr. 19**

Fișierul `atestat.in` conține o singură linie pe care se află scris un text format din cel mult 70 de caractere, primul caracter fiind literă, iar cuvintele din text sunt separate prin câte un spațiu. Fiecare cuvânt este format doar din litere mici ale alfabetului limbii engleze.

Să se scrie un program, în limbajul Pascal/C/C++, care:

- afișează textul pe o singură linie a ecranului, transformat astfel: toate vocalele vor fi scrise cu majuscule, iar restul caracterelor rămân nemodificate;
- scrie în fișierul `atestat.out`, pe un rând, separate printr-un spațiu, toate cuvintele din fișierul `atestat.in`, care se termină cu o consoană. Dacă nu există cuvinte care se termină cu o consoană, se va scrie pe prima linie a fișierului mesajul „**NU EXISTA ASTFEL DE CUVINTE**”.

**Exemplu:**

<code>atestat.in</code>
<code>examenul de bacalaureat a fost usor</code>

Date de ieșire:
a) <code>ExAmEnUl dE bAcAlAUrEAt A fOst UsOr</code>
b) Fișierul <code>atestat.out</code> conține: <code>examenul bacalaureat fost usor</code>

**Subiectul nr. 20**

Fișierul `atestat.in` conține o singură linie pe care se află scris un text format din cel mult 70 de caractere, primul caracter fiind literă, iar cuvintele din text sunt separate prin câte un spațiu. Fiecare cuvânt este format doar din litere mici ale alfabetului limbii engleze.

Să se scrie un program în limbajul Pascal/C/C++, care:

- afișează textul pe o singură linie a ecranului, transformat astfel: prima literă a fiecărui cuvânt va fi scrisă cu literă mare;
- scrie în fișierul `atestat.out`, pe prima linie, separate printr-un spațiu, toate cuvintele din fișierul `atestat.in` care încep cu o vocală. Dacă nu există cuvinte care încep cu o vocală, se va scrie pe prima linie a fișierului mesajul „**NU EXISTA ASTFEL DE CUVINTE**”.

**Exemplu:**

<code>atestat.in</code>
<code>astazi este o zi frumoasa</code>

**Date de ieșire:**

<code>a) Astazi Este O Zi Frumoasa</code>
<code>b) Fișierul atestat.out conține:</code> <code>astazi este o</code>

**Subiectul nr. 21**

Fișierul `atestat.in` conține o singură linie pe care se află scris un text format din cel mult 70 de caractere, primul caracter fiind literă, iar cuvintele din text sunt separate prin câte un spațiu. Fiecare cuvânt este format din litere mari ale alfabetului limbii engleze sau din cifre.

Să se scrie un program în limbajul Pascal/C/C++, care:

- afișează textul pe ecran, pe o singură linie, transformat astfel: toate vocalele vor fi scrise cu caractere mici, iar restul caracterelor rămân neschimbate;
- scrie în fișierul `atestat.out` pe prima linie, separate printr-un spațiu, toate cuvintele din fișierul `atestat.in` care încep cu caracterul 'c'. Dacă nu există astfel de cuvinte se va scrie pe prima linie a fișierului mesajul "NU EXISTA ASTFEL DE CUVINTE".

**Exemplu:**

<code>atestat.in</code>
MIHAI ARE 3 CREIOANE SI 5 CAIETE

Date de ieșire:
a) MiHai aRe 3 CREioaNe Si 5 Caiete
b) Fișierul <code>atestat.out</code> conține: CREIOANE CAIETE

**Subiectul nr. 22**

Fișierul `atestat.in` conține o singură linie pe care se află scris un text format din cel mult 70 de caractere, primul și ultimul caracter fiind litere, iar cuvintele din text sunt separate prin câte un spațiu. Fiecare cuvânt este format doar din litere mici ale alfabetului limbii engleze.

Să se scrie un program, în limbajul Pascal/C/C++, care:

- afișează textul pe ecran, pe o singură linie, transformat astfel: ultima literă a fiecărui cuvânt va fi scrisă cu literă mare;
- scrie în fișierul `atestat.out` pe prima linie, separate printr-un spațiu, toate cuvintele din fișierul `atestat.in` care conțin cel puțin 7 caractere. Dacă nu există astfel de cuvinte, se va scrie pe prima linie a fișierului mesajul `"NU EXISTA ASTFEL DE CUVINTE"`.

**Exemplu:**

<code>atestat.in</code>
<code>lumina soarelui se reflecta in valuri</code>

Date de ieșire:
a) <code>luminaA soareluI sE reflectA iN valurI</code>
b) Fișierul <code>atestat.out</code> conține: <code>soarelui reflecta</code>

**Subiectul nr. 23**

Fișierul `atestat.in` conține pe prima linie numărul natural nenul  $n$  ( $2 < n < 10$ ). Pe următoarele  $n$  linii sunt scrise câte două numere naturale  $x$  și  $y$  ( $x < y$ ), reprezentând capetele unor intervale. Cele două numere naturale  $x$  și  $y$  sunt separate printr-un spațiu.

Utilizând structuri de date tip înregistrare să se scrie un program, în limbajul Pascal/C/C++, care:

- afișează pe câte o linie a ecranului, cele  $n$  intervale sub forma `[x , y]` ([capăt stânga, capăt dreapta]);
- scrie în fișierul `atestat.out`, pe prima linie, separate printr-un spațiu, capetele intervalului de intersecție a tuturor celor  $n$  intervale. Dacă nu există un astfel de interval se va afișa mesajul "NU EXISTA INTERVAL DE INTERSECȚIE".

**Exemplu:**

<code>atestat.in</code>	Date de ieșire:
3	a) [10,40]
10 40	[5,20]
5 20	[6,30]
6 30	b) Fișierul <code>atestat.out</code> conține: 10 20

**Subiectul nr. 24**

Fișierul `atestat.in` conține pe prima linie numărul natural nenul  $n$  ( $2 < n < 10$ ). Pe următoarele  $n$  linii sunt scrise câte două numere naturale nenule  $c$  și  $p$ , reprezentând **codul** și **prețul** obiectelor unei case de licitație. Cele două numere  $c$  și  $p$  sunt separate printr-un spațiu.

Utilizând structuri de date tip înregistrare să se scrie un program, în limbajul Pascal/C/C++, care:

- afișează în linie pe ecran, separate prin câte un spațiu, cele  $n$  obiecte sub forma  $(c, p)$  (**cod obiect, preț obiect**);
- scrie în fișierul `atestat.out`, pe câte un rând, codul obiectului/ obiectelor cu preț minim.

**Exemplu:**

<code>atestat.in</code>	Date de ieșire:
6	a) (3,20) (5,10) (1,40) (4,10) (2,70) (6,30)
3 20	b) Fișierul <code>atestat.out</code> conține:
5 10	5
1 40	4
4 10	
2 70	
6 30	



**Subiectul nr. 25**

Fișierul `atestat.in` conține pe prima linie numărul natural nenul  $n$  ( $2 < n < 10$ ). Pe fiecare dintre următoarele  $n$  linii sunt scrise câte două numere naturale  $x$  și  $y$ , reprezentând coordonatele carteziene ale unui punct din planul  $xOy$ . Cele două numere  $x$  și  $y$  sunt separate printr-un spațiu.

Utilizând structuri de date tip înregistrare să se scrie un program, în limbajul Pascal/C/C++, care:

- afișează pe ecran punctele din planul  $xOy$ , sub forma  $(x,y)$  (abscisă, ordonată) câte unul pe linie;
- scrie în fișierul `atestat.out`, câte puncte din fișierul `atestat.in` sunt situate pe *prima bisectoare*. Un punct se află situat pe *prima bisectoare* atunci când abscisa și ordonata acestuia sunt egale.

**Exemplu:**

<code>atestat.in</code>	Date de ieșire:
5	a) (1,3)
1 3	(4,4)
4 4	(2,5)
2 5	(3,3)
3 3	(6,2)
6 2	b) Fișierul <code>atestat.out</code> conține: 2