**Coordonator:** prof. Gabriela-Rodica Freitag

**Autori:**

prof. Costineanu Raluca-Veronica

prof. Freitag Gabriela-Rodica

prof. Hrișcă Silvia-Steluța

prof. Ildegez Mihaela-Corina

prof. Ilincăi Florin

prof. Petrișor  Adrian

prof. Purice Anca-Cristina

prof. Ștefănescu Narcisa-Daniela

prof. Ududec Marius

Examen de atestare a competențelor profesionale ale absolvenților claselor de matematică-informatică și matematică-informatică, intensiv informatică, în anul școlar 2025-2026

# Examen de atestare a competențelor profesionale ale absolvenților claselor de matematică-informatică și matematică-informatică, intensiv informatică, în anul școlar 2025-2026

## Subiecte PROGRAMARE

### SUBIECTUL NR. 1

**Enunţ**

Fişierul *atestat.in* conţine pe prima linie un număr natural **n** (2≤n≤100), iar pe a doua linie, un şir de **n** numere naturale cu cel puţin 2 cifre şi cel mult 9 cifre fiecare, separate prin câte un spaţiu.

**a)** Funcţia *primaCifra* determină şi returnează prima cifră a unui număr natural **x,** cu cel mult 9 cifre, primit ca parametru. Scrieţi funcţia *primaCifra* definită mai sus.

**b)** Folosind apeluri utile ale subprogramului *de la punctul a,* determinaţi şi afişaţi pe ecran câte numere citite din fişier au prima cifră egală cu ultima. Dacă nu există niciun număr cu această proprietate, atunci se va scrie *Nu există!*.

**Exemplu**

| **Date de intrare**<br>***atestat.in*** | **Date de ieşire** |
|---|---|
| 7<br>1421 237 7447 727 2321 88 90 | Sunt 4 numere. |

### SUBIECTUL NR. 2

**Enunţ**

Fişierul *atestat.in* conţine pe prima linie un număr natural **n** (2≤n≤100), iar pe a doua linie, un şir de **n** numere naturale cu cel mult 9 cifre fiecare, separate prin câte un spaţiu.

**a)** Funcţia *primaCifra* determină şi returnează prima cifră a unui număr natural **x,** cu cel mult 9 cifre**,** primit ca parametru. Scrieţi funcţia *primaCifra* definită mai sus.

**b)** Folosind apeluri utile ale subprogramului definit *la punctul a,* determinaţi dacă numărul format din prima cifră a fiecărui număr din fişier este divizibil cu numărul 3.

**Exemplu**

| **Date de intrare**<br>***atestat.in*** | **Date de ieşire** |
|---|---|
| 7<br>12 34 567 78 988 3333 289 | Numărul 1357932 este divizibil cu 3. |

## SUBIECTUL NR. 3

**Enunţ**

Fişierul *atestat.in* conţine pe prima linie un număr natural **n** (2≤n≤100), iar pe a doua linie, un şir de **n** numere naturale cu cel puţin 2 cifre şi cel mult 9 cifre fiecare, separate prin câte un spaţiu.

**a)** Funcţia *oglinditPar* determină şi returnează oglinditul format doar din cifrele pare ale unui număr natural x (cu cel mult 9 cifre), primit ca parametru. Scrieţi definiţia funcţiei.

**b)** Folosind apeluri utile ale subprogramului definit *la punctul a,* determinaţi şi afişaţi câte elemente din şir (exclusiv primul număr) au oglinditul par egal cu oglinditul par al primului element din şir. Dacă nu există niciun număr cu această proprietate, atunci se va scrie *Nu există!*.

**Exemplu**

| **Date de intrare**<br>***atestat.in*** | **Date de ieşire** |
|---|---|
| 10<br>2145 7897 2345 287 654 7294 425 6854 365 240 | Sunt 3 numere. |

**Explicaţie:** numerele 2345 7294 240 au oglinditul par egal cu al lui 2145.

## SUBIECTUL NR. 4

**Enunţ**

Fişierul *atestat.in* conţine pe prima linie un număr natural **n** (2≤n≤100), iar pe următoarele **n** linii, **n** perechi de numere naturale cu cel puţin 2 cifre şi cel mult 9 cifre fiecare, separate prin câte un spaţiu.

**a)** Funcţia *sumaDiv* determină şi returnează suma divizorilor unui număr natural x (cu cel mult 9 cifre), primit ca parametru. Scrieţi definiţia funcţiei.

**b)** Folosind apeluri utile ale subprogramului definit *la punctul a,* determinaţi şi afişaţi perechile de numere de pe aceeaşi linie care sunt quasi-amiabile. Două

numere naturale cu proprietatea că suma divizorilor fiecărui număr (neconsiderând în acest caz numărul însuși ca divizor) este egală cu celălalt număr plus 1, se numesc numere quasi-amiabile sau numere logodite. Dacă nu există nicio astfel de pereche, se afișează mesajul *Nu există!*.

**Exemplu**

| **Date de intrare**<br>***atestat.in*** | **Date de ieşire** |
|---|---|
| 4<br>25 18<br>48 75<br>195 140<br>260 310 | 48 75<br>195 140 |

## SUBIECTUL NR. 5

**Enunţ**

Fişierul *atestat.in* conţine pe prima linie un număr natural **n** (2≤n≤100), iar pe următoarele **n** linii, **n** numere naturale cu cel mult 9 cifre fiecare, separate prin câte un spaţiu.

**a)** Scrieţi definiţia completă a unui subprogram *baza1*, care primeşte prin intermediul parametrului **x** un număr natural nenul de cel mult 9 cifre; subprogramul returnează numărul de cifre de 1 din reprezentarea în baza 2 a numărului **x**. Identic funcţia *baza0* returnează numărul de cifre de 0 din reprezentarea în baza 2 a numărului **x**.

**b)** Scrieţi programul **C++** care afişează pe ecran, separate prin câte un spaţiu, toate numerele din fişierul text *atestat.in* care sunt rotunde. Dacă nu există niciun număr rotund, atunci se va scrie pe ecran *Nu există!*.

**Explicaţie:** Un număr **n** se numeşte **rotund** dacă are un număr egal de cifre de 0 şi 1 în reprezentarea sa matematică în baza 2. De exemplu, numărul 184 este un număr rotund deoarece reprezentat în baza 2 conţine 4 cifre de 0 şi 4 cifre de 1. Numărul 120 nu este rotund deoarece reprezentat în baza 2 conţine 3 cifre de 0 şi 4 cifre de 1.

**Exemplu**

| **Date de intrare**<br>***atestat.in*** | **Date de ieşire** |
|---|---|
| 8<br>21 142 10 45 15 7 62 216 | 142 10 216 |
| 8<br>43 214 7 68 144 302 24 2140 | Nu exista! |

## SUBIECTUL NR. 6

**Enunţ**

Fişierul *atestat.in* conţine pe prima linie un număr natural **n** (2≤n≤100), iar pe următoarele **n** linii, **n** numere naturale cu cel mult 9 cifre fiecare, separate prin câte un spaţiu.

**a)** Scrieţi definiţia completă a unui subprogram *cifraControl*, cu un parametru care primeşte prin intermediul parametrului **x**, un număr natural nenul de cel mult 9 cifre; subprogramul returnează cifra de control a numărului **x**.

**Exemplu:** pentru **n=8488** subprogramul va returna valoare 1(suma cifrelor lui 8488 este 28, suma cifrelor lui 28 este 10, suma cifrelor lui 10 este 1), iar pentru **n=2015** subprogramul va returna valoarea 8 (suma cifrelor lui 2015 este 8).

**b)** Scrieţi programul **C++** care afişează pe ecran care este numărul de elemente din fişierul de intrare care au cifra de control număr prim. Dacă nu există niciun număr cu cifra de control număr prim, atunci se va scrie *Nu există!*.

**Explicaţie: Cifra de control** a unui număr natural **n** se obţine efectuând suma cifrelor numărului, apoi suma cifrelor sumei obţinute şi aşa mai departe până când se obţine un număr format dintr-o singură cifră.

**Exemplu**

| **Date de intrare**<br>***atestat.in*** | **Date de ieşire** |
|---|---|

| | |
|---|---|
| 8<br>124 54 9199 5 8712 230 19 7778 | Sunt 4 numere<br>. |

## SUBIECTUL NR. 7

**Enunţ**

Fişierul text *atestat.in* conţine, pe prima linie un număr natural **n** (n≤100), iar pe următoarele **n** linii, fracţii de forma $\frac{a}{b}$, unde **a** şi **b** sunt numărătorul, respectiv numitorul, numere naturale nenule de cel mult patru cifre fiecare, separate prin câte un spaţiu.

**a)** Scrieţi definiţia completă a unui subprogram *cmmdc*, cu doi parametri, care primeşte prin intermediul parametrilor **x** şi **y** două numere naturale nenule de cel mult patru cifre fiecare. Subprogramul returnează cel mai mare divizor comun al celor două numere **x** şi **y**.

**b)** Scrieţi programul **C++** care citeşte şi simplifică fracţiile din fişierul de intrare şi care afişează pe ecran numărătorul şi numitorul fracţiei ireductibile obţinute după simplificare, separate prin câte un spaţiu.

**Exemplu**

| **Date de intrare**<br>***atestat.in*** | **Date de ieşire** |
|---|---|
| 4<br>26 12<br>19 4<br>20 15<br>17 17 | 13 6<br>19 4<br>4 3<br>1 1 |

## SUBIECTUL NR. 8

**Enunţ**

Fişierul *atestat.in* conţine pe prima linie un număr natural **n** (2≤n≤100), iar pe următoarele **n** linii, **n** numere naturale cu cel mult 9 cifre fiecare, separate prin câte un spaţiu.

**a)** Scrieţi definiţia completă a unui subprogram cu numele *puternic*, care primeşte prin intermediul parametrului **x,** un număr natural de cel mult 9 cifre. Subprogramul returnează true dacă numărul **x** este puternic şi false în caz contrar.

**b)** Folosind apeluri utile ale subprogramului *de la punctul a,* afișați pe ecran câte numere puternice sunt în șir. Dacă nu există niciun număr puternic atunci se va scrie Nu există*!*.

**Explicaţie**: Un număr n este număr puternic dacă numărul se poate scrie ca sumă a cifrelor ridicate la puterea numărului de cifre ale lui n. Exemple de astfel de numere: $371=3^3+7^3+1^3$ , $1634=1^4+6^4+3^4+4^4$

**Exemplu**

| **Date de intrare**<br>**atestat.in** | **Date de ieşire** |
|---|---|
| 10<br>0 237 1447 371 1271 1634 407 1023<br>120 9474 | Sunt 5 numere puternice. |

## SUBIECTUL NR. 9

**Enunţ**

a) Scrieţi definiţia completă a unui subprogram cu numele *cifraMax*, cu un parametru, care primeşte prin intermediul parametrului **x** un număr natural de cel mult 9 cifre. Subprogramul returnează cifra maximă din număr.

b) Folosind apeluri utile ale subprogramului *de la punctul a)* afișați pe ecran câte numere din șir au cifra maximă egală cu a ultimului element.

**Exemplu**

| **Date de intrare** | **Date de ieşire** |
|---|---|

| ***atestat.in*** | |
|---|---|
| 7<br>37 40 43 60 31 11 140 | Sunt 3 numere care au cifra maximă egală cu 4. |

## SUBIECTUL NR. 10

**Enunţ**

Prin înjumătăţirea cifrelor unui număr natural se înţelege înlocuirea fiecărei cifre cu câtul împărţirii cifrei la 2. De exemplu, din numărul 51324 prin înjumătăţirea cifrelor se va obţine 20112. Fişierul atestat.in conţine pe prima linie un număr natural **n** (2≤n≤100), iar pe a doua linie, un şir de **n** numere naturale cu cel mult 9 cifre fiecare, separate prin spaţiu.

a) Să se scrie definiţia completă a subprogramului *InjumatatireCifre* care are ca parametru numărul natural **x**. Subprogramul înjumătăţeşte fiecare cifră a numărului **x**, conform definiţiei de mai sus şi returnează numărul modificat.

b) Să se scrie un program care citeşte din fişierul atestat.in numărul **n** şi cele **n** numere şi folosind apeluri utile ale subprogramului definit *la punctul a*, afişează pe ecran câte dintre cele **n** numere au toate cifrele pare. Dacă nu există niciun număr cu această proprietate, atunci se va scrie *Nu există!*.

**Exemplu**

| **Date de intrare**<br>**atestat.in** | **Date de ieşire** |
|---|---|
| 7<br>634 8644 1024 102 2048 25684 2083 | Sunt 2 numere. |

## SUBIECTUL NR. 11

**Enunţ**

Din fişierul text *atestat.in* se citeşte un număr natural **n** (**n≤100**) reprezentând numărul de linii şi numărul de coloane ale unui tablou bidimensional. De pe următoarele **n** linii se citesc elementele tabloului, câte **n** componente de pe fiecare linie, numere naturale nenule cu cel mult patru cifre fiecare.

**a)** Scrieţi definiţia completă a unui subprogram cu numele *elementMax*, cu un parametru, care primeşte prin intermediul parametrului **x** un număr natural **x≤n**. Subprogramul returnează elementul maxim de pe linia x a matricei.

**b)** Folosind apeluri utile ale subprogramului definit *la punctul a,* determinaţi şi afişaţi câte elemente maxime pe linia lor se găsesc pe diagonală şi sub diagonala principală a matricei.

**Exemplu**

| **Date de intrare**<br>***atestat.in*** | **Date de ieşire** |
|---|---|
| 4<br>1 2 1 3<br>43 454 234 12<br>23 77 99 1<br>8 6 5 3 | Sunt 3 maxime. |

## SUBIECTUL NR. 12

**Enunţ**

Din fişierul text *atestat.in* se citeşte un număr natural **n** (**n≤100**) reprezentând numărul de linii şi numărul de coloane ale unui tablou bidimensional. De pe următoarele **n** linii se citesc elementele tabloului, câte **n** componente de pe fiecare linie, numere naturale nenule cu cel mult 9 cifre fiecare.

**a)** Funcţia *numarPrim* verifică dacă un număr natural **x,** cu cel mult 9 cifre**,** primit ca parametru este număr prim şi returnează true (sau valoarea 1) în caz afirmativ,

respectiv false (sau valoarea 0) în caz contrar. Scrieţi funcţia *numarPrim* definită mai sus.

**b)** Folosind apeluri utile ale subprogramului *de la punctul a,* determinaţi şi afişaţi pe ecran numerele aflate pe conturul matricei care sunt numere prime. Ordinea în care sunt afişate numerele nu este importantă, iar dacă nu există niciun număr cu această proprietate, atunci se va scrie Nu există!

**Exemplu**

| **Date de intrare**<br><br>***atestat.in*** | **Date de ieşire** |
|---|---|
| 4<br>124 564 871 117<br>537 224 177 677<br>111 235 657 589<br>711 211 677 671 | 677 211 677 |

## SUBIECTUL NR. 13

**Enunţ**

Din fişierul text *atestat.in* se citesc: un număr natural **n** (**n≤100**) reprezentând numărul de linii şi numărul de coloane ale unui tablou bidimensional. De pe următoarele **n** linii se citesc elementele tabloului, câte **n** componente de pe fiecare linie, numere naturale nenule cu cel mult 9 cifre fiecare.

**a)** Funcţia *numarPalindrom* verifică dacă un număr natural **x,** cu cel mult 9 cifre**,** primit ca parametru este număr palindrom şi returnează true (sau valoarea 1) în caz afirmativ, respectiv false (sau valoarea 0) în caz contrar. Scrieţi funcţia *numarPalindrom* definită mai sus.

**b)** Folosind apeluri utile ale subprogramului *de la punctul a,* determinaţi şi afişaţi pe ecran numerele palindrom aflate pe diagonala principală sau pe diagonala secundară

a matricei. Se va afişa mesajul "Nu există" în cazul în care nici una din cele două diagonale nu conţin numere palindrom.

**Exemplu**

| **Date de intrare**<br>***atestat.in*** | **Date de ieşire** |
|---|---|
| 5<br>121 34 565 787 2<br>3 4 5 677 6<br>45 67 21 9 8<br>6 44 21 3 55<br>78 90 545 2 15 | 121 4 3 2 44 |

## SUBIECTUL NR. 14

**Enunţ**

Din fişierul text *atestat.in* se citesc: numerele naturale **n** şi **m** (**n,m≤100**) reprezentând numărul de linii, respectiv numărul de coloane ale unui tablou bidimensional. De pe următoarele **n** linii se citesc elementele tabloului, câte **m** componente de pe fiecare linie, numere naturale nenule cu cel mult patru cifre fiecare.

**a)** Scrieţi definiţia completă a unui subprogram cu numele *sumaColoana*, cu un parametru, care primeşte prin intermediul parametrului **x** un număr natural **x≤m**. Subprogramul returnează suma elementelor de pe coloana x a matricei.

**b)** Folosind apeluri utile ale subprogramului definit *la punctul a,* să se afişeze suma maximă obţinută şi coloana pe care s-a obţinut. Dacă există mai multe coloane de sumă maximă se va afişa cel mai mic dintre indicii acestora.

**Exemplu**

| **Date de intrare**<br>***atestat.in*** | **Date de ieşire** |
|---|---|
| 4 5<br>1 2 3 4 3<br>4 5 6 5 3<br>7 8 9 6 9<br>3 1 8 5 1 | Suma maxima este 26 si s-a obtinut pe coloana 3. |

## SUBIECTUL NR. 15

**Enunţ**

Din fişierul text *atestat.in* se citesc: un număr natural **n** (**n≤100**) reprezentând numărul de linii şi numărul de coloane ale unui tablou bidimensional. De pe următoarele linii se citesc elementele tabloului, câte **n** componente de pe fiecare linie, numere naturale nenule cu cel mult 9 cifre fiecare.

**a)** Funcţia *numarPerfect* verifică dacă un număr natural **x,** cu cel mult 9 cifre**,** primit ca parametru este număr perfect şi returnează true (sau valoarea 1) în caz afirmativ, respectiv false (sau valoarea 0) în caz contrar. Scrieţi funcţia *numarPerfect* definită mai sus.

**b)** Folosind apeluri utile ale subprogramului *de la punctul a,* determinaţi şi afişaţi pe ecran numerele perfecte aflate pe diagonala secundară a matricei. Se va afişa mesajul "Nu există" în cazul în care diagonala secundară nu conţine numere perfecte.

**Explicaţie:** Un număr este perfect dacă este egal cu suma divizorilor săi proprii pozitivi plus 1.

**Exemplu**

| **Date de intrare**<br>***atestat.in*** | **Date de ieşire** |
|---|---|
| 4<br>1 2 16 6 | 6 28 |

| | |
|---|---|
| 4 56 32 8<br>11 28 14 78<br>15 13 9 80 | |

## SUBIECTUL NR. 16

**Enunţ**

Din fişierul text *atestat.in* se citeşte un număr natural **n** (**n≤100**) reprezentând numărul de linii şi numărul de coloane ale unui tablou bidimensional. De pe următoarele linii se citesc elementele tabloului, câte **n** componente de pe fiecare linie, numere naturale nenule cu cel mult patru cifre fiecare.

**a)** Funcţia *nrDiv* determină şi returnează pentru un număr natural **x,** cu cel mult 9 cifre**,** primit ca parametru numărul de divizori proprii. Scrieţi funcţia *nrDiv* definită mai sus.

**b)** Folosind apeluri utile ale subprogramului *de la punctul a,* determinaţi şi afişaţi pe ecran numerele aflate sub diagonala secundară care au număr impar de divizori. Se va afişa mesajul "Nu există" în cazul în care sub diagonala secundară nu există numere cu proprietatea cerută.

**Exemplu**

| **Date de intrare**<br>***atestat.in*** | **Date de ieşire** |
|---|---|
| 4<br>12 4 19 8<br>2 3 5 4<br>1 10 1 18<br>9 0 9 11 | 4 1 9 |

## SUBIECTUL NR. 17

**Enunţ**

Din fişierul text *atestat.in* se citesc: două numere naturale, **n şi m** (**n,m≤100**) reprezentând numărul de linii şi numărul de coloane ale unui tablou bidimensional. De pe următoarele linii se citesc elementele tabloului, câte **m** numere naturale cu cel mult 9 cifre fiecare de pe fiecare linie a tabloului. Numerotarea liniilor şi a coloanelor se face începând cu valoarea 1.

**a)** Funcţia *numarZero* determină şi returnează numărul de valori de zero de pe linia x, 1≤x≤n **,** primit ca parametru. Scrieţi funcţia *numarZero* definită mai sus.

**b)** Folosind apeluri utile ale subprogramului *de la punctul a,* determinaţi şi afişaţi pe ecran indicii liniilor care nu au niciun element egal cu valoarea 0. Se va afişa mesajul "Nu există" în cazul în care toate liniile conţin cel puţin o valoare egală cu 0.

**Exemplu**

| **Date de intrare**<br>***atestat.in*** | **Date de ieşire** |
|---|---|
| 4 5<br>2 3 5 4 10<br>1 0 11 18 78<br>9 1 9 11 345<br>2 2 45 7 0 | 1 3 |

## SUBIECTUL NR. 18

**Enunţ**

Din fişierul text *atestat.in* se citesc două numere naturale, n şi m (**n,m≤100**) reprezentând numărul de linii şi numărul de coloane ale unui tablou bidimensional. De pe următoarele linii se citesc elementele tabloului, câte **m** componente de pe fiecare linie a tabloului, numere naturale din mulţimea {0,1}.

Într-un tablou bidimensional, cu elemente având valori numai din mulţimea {0,1}, numim coloane "***complementare***" două coloane cu proprietatea că oricare două elemente ale acestora, aflate pe aceeaşi linie, sunt diferite.

**a)** Scrieţi definiţia completă a subprogramului *complementare* care primeşte ca parametrii două numere naturale c1 şi c2, reprezentând indicii a două coloane din matrice. Subprogramul returnează true (sau valoarea 1) dacă coloanele c1 şi c2 sunt complementare, respectiv false (sau valoarea 0) în caz contrar.

**b)** Folosind apeluri utile ale *subprogramului de la punctul a*, determinaţi şi afişaţi pe ecran numărul de coloane ale tabloului care sunt complementare cu prima coloană a acestuia.

**Exemplu**

| **Date de intrare**<br>***atestat.in*** | **Date de ieşire** |
|---|---|
| 3 6<br>1 1 0 0 1 0<br>0 1 1 1 1 1<br>1 0 0 0 0 1 | Sunt 2 coloane. |

## SUBIECTUL NR. 19

**Enunţ**

Din fişierul text *atestat.in* se citesc se citește un număr natural **n**, apoi câte **n** numere naturale nenule cu cel mult 9 cifre fiecare de pe fiecare linie.

**a)** Funcţia *divizorMax* determină şi returnează divizorul prim cel mai mare al unui număr x, cu cel mult 9 cifre**,** primit ca parametru. Scrieţi funcţia *divizorMax* definită mai sus.

**b)** Folosind apeluri utile ale subprogramului *de la punctul a,* determinaţi şi afişaţi pe ecran numerele prime aflate pe una din cele două diagonale ale matricei. Se va afişa

mesajul "Nu există" în cazul în care nici una din cele două diagonale nu conţine numere prime.

**Exemplu**

| **Date de intrare**<br>***atestat.in*** | **Date de ieşire** |
|---|---|
| 4<br>2 1 6 4<br>3 1 12 9<br>10 11 13 15<br>23 41 52 18 | 2 13 11 23 |

## SUBIECTUL NR. 20

**Enunţ**

Din fişierul text *atestat.in* se citeşte un număr natural **n** (**n≤100**) reprezentând numărul de linii şi numărul de coloane ale unui tablou bidimensional. De pe următoarele linii se citesc elementele tabloului, câte **n** numere naturale nenule cu cel mult 9 cifre fiecare, de pe fiecare linie a tabloului. De pe ultima linie a fişierului se citeşte un număr natural nenul k, k<=9.

**a)** Funcţia *numarAscuns* primeşte doi parametri, un număr natural **x,** cu cel mult 9 cifre şi un număr natural k, **k≤9** . Ultimele k cifre ale numărului **x** vor fi înlocuite cu zero. Scrieţi funcţia *numarAscuns* definită mai sus.

**b)** Folosind apeluri utile ale subprogramului *de la punctul a,* determinaţi şi afişaţi pe ecran numerele aflate pe ultimele două linii a matricei care nu sunt modificate după transformare.

## SUBIECTUL NR. 21

**Enunţ**

Din fişierul text *atestat.in* se citește un cuvânt de maxim 100 de caractere, care este alcătuit doar din litere mici ale alfabetului englez.

**a)** Funcţia *sirPalindrom* primește ca parametru de intrare un șir de maxim 100 de caractere și returnează valoarea 1, dacă șirul are proprietate de palindrom (citit de la stânga la dreapta este egal cu șirul citit de la dreapta la stânga) sau 0 în caz contrar.

**b)** Folosind apeluri utile ale subprogramului *de la punctul a,* determinaţi şi afişaţi pe ecran **cel mai lung prefix** și **cel mai lung sufix** cu **proprietate de palindrom** care se poate obține din cuvântul s. Cele două cuvinte se vor afișa pe o singură linie separate printr-un spațiu.

**Exemplu**

| **Date de intrare**<br>**atestat.in** | **Date de ieşire** |
|---|---|
| `anastasia` | `ana a` |
| `capac` | `capac capac` |

## SUBIECTUL NR. 22

**Enunţ**

Din fişierul text *atestat.in* citește un șir de maxim 255 de caractere, litere mici ale alfabetului englez și spații. Cuvintele din text sunt separate printr-un singur spațiu, iar în fața primului și ultimului cuvânt nu există spații.

**a)** Funcţia *nrConsoane* primește ca parametru de intrare un șir de maxim 30 de caractere și returnează **numărul de consoane** din șir.

**b)** Folosind apeluri utile ale subprogramului *de la punctul a,* modificați în memorie șirul dat eliminând **cuvintele care conțin un număr de consoane egal cu numărul de vocale.** Șirul final va fi afișat pe ecran, dacă s-a eliminat cel puțin un cuvânt, iar dacă șirul inițial nu conține nici un cuvânt care aibă numărul de vocale egal cu numărul de consoane se va afișa pe ecran mesajul **nemodificat**.

**Exemplu**

| **Date de intrare**<br>***atestat.in*** | **Date de ieşire** |
|---|---|

| | |
|---|---|
| este un an nou cu bucurii si implinirea dorintelor | nou bucurii dorintelor |
| tarca e rea. | nemodificat |

## SUBIECTUL NR. 23

**Enunț**

Din fișierul text *atestat.in* citește un șir de maxim 255 de caractere, litere mici ale alfabetului englez și spații. Cuvintele din text sunt separate printr-un singur spațiu iar în fața primului și ultimului cuvânt nu există spații.

**a)** Funcția *nrVocale* primește ca parametru de intrare un șir de maxim 100 de caractere și returnează **numărul de vocale** din șir.

**b)** Folosind apeluri utile ale subprogramului *de la punctul a,* afișați cuvântul (cuvintele) cu cele mai multe vocale. Dacă există mai multe cuvinte, se vor afișa separate printr-un spațiu.

**Exemplu**

| **Date de intrare**<br>***atestat.in*** | **Date de ieșire** |
|---|---|
| vara aceasta matei si andreea au fost la bunici | aceasta andreea |

## SUBIECTUL NR. 24

**Enunț**

Din fișierul text *atestat.in* citește un șir de maxim 255 de caractere, litere mici ale alfabetului englez și spații. Cuvintele din text sunt separate printr-un singur spațiu iar în fața primului și ultimului cuvânt nu există spații.

**a)** Funcția *cautLitera* primește ca parametru de intrare un șir de maxim 100 de caractere și o literă c și returnează valoarea 1 dacă litera se găsește în șir sau 0 în caz contrar.

**b)** Folosind apeluri utile ale subprogramului de la punctul a, afișați cuvântul (cuvintele) care conțin litera cu care începe primul cuvânt din șir. Primul cuvânt nu se va afișa.

Dacă în şirul dat nu există cuvinte cu proprietatea cerută se va afişa mesajul "Nu există".

**Exemplu**

| **Date de intrare**<br>***atestat.in*** | **Date de ieşire** |
|---|---|
| anul acesta voi termina studiile liceale | acesta termina liceale |

## SUBIECTUL NR. 25

**Enunţ**

Din fişierul text *atestat.in* citește un șir de maxim 255 de caractere, litere mici ale alfabetului englez și spații. Cuvintele din text sunt separate printr-un singur spațiu iar în fața primului și ultimului cuvânt nu există spații.

**a)** Funcţia *schimbLitera* primește ca parametru de intrare o literă c și o transformă în literă mare și o returnează.

**b)** Folosind apeluri utile ale subprogramului de la punctul a, afişaţi pe ecran şirul transformat, astfel încât fiecare cuvânt va începe cu literă mare.

**Exemplu**

| **Date de intrare**<br>***atestat.in*** | **Date de ieşire** |
|---|---|
| voi merge la bunici la vara | Voi Merge La Bunici La Vara |