

## SUBIECTE ATESTAT - PROGRAMARE

1. Se citesc  $n$  numere naturale, unde  $n$  este un număr natural nenul dat.
  - a. Să se afișeze valoarea minimă și valoarea maximă din șir, precum și pozițiile pe care se regăsesc acestea.
  - b. Să se afișeze cele mai mari două valori din șir.
2. Se citesc  $n$  numere naturale  $a_1, a_2, \dots, a_n$ ,  $n$  dat,  $0 < n < 100$ .
  - a. Să se afișeze numărul de numere prime din șir.
  - b. Să se determine numărul cifrelor 0 în care se termină numărul natural  $p = a_1 \times a_2 \times \dots \times a_n$ .
3. În fișierul **atestat.in** se găsesc, pe prima linie, un număr natural  $n$ , nenul și pe a doua linie un șir de  $n$  numere naturale, despărțite de câte un spațiu.
  - a. Să se afișeze cifra care apare de cele mai multe ori în scrierea numerelor citite. Dacă sunt mai multe astfel de cifre, se va afișa cea mai mare dintre ele.
  - b. Să se afișeze, pe o singură linie, despărțite prin câte un spațiu, toate cifrele care nu apar în scrierea niciunui număr din șir.
4. Se citesc din fișierul **atestat.in** două numere naturale  $n$  și  $k$ . Să se afișeze, în fișierul **atestat.out**,
  - a. toate puterile numărului  $k$  mai mici sau egale cu  $n$ . (Exemplu: dacă  $n=10$  și  $k=2$ , se va afișa 1,2,4,8) Numere vor fi afișate pe o linie, cu virgulă între ele.
  - b. În cazul în care  $k$  este număr prim, se va afișa, pe linia următoare din fișier, puterea la care apare  $k$ , în descompunerea lui  $n$  în factori primi. În cazul în care  $k$  nu este prim, se va afișa puterea la care apare 2 în descompunerea în factori primi a lui  $k$ .
5. În fișierul **atestat.in** se găsesc, pe prima linie, un număr natural  $n$ , nenul și pe a doua linie un șir de  $n$  numere întregi, despărțite de câte un spațiu. Să se afișeze, în fișierul **atestat.out**:
  - a. acele numere din șir care se divid cu suma cifrelor lor. Numere vor fi afișate pe o singură linie, cu spațiu între ele.
  - b. pe linia următoare din fișier, se va afișa cel mai mic multiplu comun al numerelor din șir.
6. În fișierul **atestat.in** se găsesc  $n$  perechi de numere naturale de forma  $(a, b)$ , scrise pe câte o linie, cu spațiu între elementele perechii. Să se afișeze, în fișierul **atestat.out**,
  - a. numărătorul și numitorul fracției ireductibile la care poate fi adusă fracția  $a/b$ .
  - b. numărul de fracții care erau deja ireductibile.
7. Se citește din fișierul **atestat.in**, un număr natural  $n$ . Să se afișeze, în fișierul **atestat.out**:
  - a. al  $n$ -lea termen din sirul lui Fibonacci, folosind o funcție care returnează al  $n$ -lea element din acest șir.
  - b. valoarea lui  $n$ , în baza 2. Valoarea va fi scrisă pe linia următoare din fișier.

8. În fișierul **atestat.in** se găsesc, pe prima linie, un număr natural  $n$ , nenul și pe a doua linie un șir de  $n$  cifre, despărțite de câte un spațiu. Să se afișeze, în fișierul **atestat.out**:
- pe prima linie, cel mai mare număr ce poate fi obținut cu cifrele numărului inițial.
  - pe a doua linie, separate prin câte un spațiu, toate numerele prime ce pot fi scrise folosind cifre din fișier. Fiecare cifră din fișier, se va folosi în scrierea numerelor prime, de un număr de ori mai mic sau egal cu numărul de apariții a sa în fișierul de intrare.
9. În fișierul **atestat.in** se găsesc, pe prima linie, un număr natural  $n$ , nenul și pe a doua linie un șir de  $n$  numere întregi, despărțite de câte un spațiu. Să se afișeze, în fișierul **atestat.out**:
- pe prima linie, toate numerele impare din fișier în ordine descrescătoare, separate prin câte un spațiu.
  - pe a doua linie din fișier, toate numerele din șir în ordine crescătoare, cele pare rămânând pe loc. (Ex. Dacă  $n=5$  și  $v=(10,7,3,2,4)$ , se va obține  $v=(10,3,7,2,4)$ ).
10. În fișierul **atestat.in** se găsesc, pe prima linie, un număr natural  $n$ , nenul și pe a doua linie un șir de  $n$  numere întregi, despărțite de câte un spațiu. Să se afișeze, în fișierul **atestat.out**:
- pe prima linie, ordonate crescător. toate elementele vectorului aflate între poziția maximului și poziția minimului, inclusiv acestea.
  - pe a doua linie, elementele vectorului după ștergerea tuturor aparițiilor primului element în vector.
11. Se dă o matrice pătratică cu  $n$  elemente numere întregi. Matricea se citește din fișierul **matrice.in**, de pe prima linie numărul  $n$ , iar de pe următoarele  $n$  linii, valorile de pe fiecare linie a matricei. În fișierul **matrice.out** se vor afișa, pe prima, respectiv a doua linie, următoarele valori:
- Media aritmetică a elementelor strict pozitive de deasupra diagonalei principale;
  - Numărul elementelor negative de sub diagonala principală.
12. Se dă o matrice pătratică cu  $n$  elemente numere întregi. Matricea se citește din fișierul **matrice.in**, de pe prima linie numărul  $n$ , iar de pe următoarele  $n$  linii, valorile de pe fiecare linie a matricei. În fișierul **matrice.out** se vor afișa, pe primele  $n$  linii, respectiv pe următoarele  $n$  linii, următoarele valori:
- Matricea obținută în urma inversării liniei  $L1$  cu linia  $L2$  din matricea dată;
  - Matricea obținută în urma inversării coloanei  $C1$  cu coloana  $C2$  din matricea inițială.
13. Se dă o matrice pătratică cu  $n$  elemente numere întregi. Matricea se citește din fișierul **matrice.in**, de pe prima linie numărul  $n$ , iar de pe următoarele  $n$  linii, valorile de pe fiecare linie a matricei. În fișierul **matrice.out** se vor afișa, pe prima, respectiv a doua linie, valorile DA sau NU, în funcție de rezultatul următoarelor verificări:
- Dacă matricea este simetrică față de diagonala principală, se va afișa DA, altfel NU;
  - Dacă matricea este simetrică față de diagonala secundară, se va afișa DA, altfel NU.

14. Se dă o matrice pătratică cu  $n$  elemente numere întregi. Matricea se citește din fișierul **matrice.in**, de pe prima linie numărul  $n$ , iar de pe următoarele  $n$  linii, valorile de pe fiecare linie a matricei. În fișierul **matrice.out** se vor afișa următoarele valori:

- a. Pe prima linie se va afișa produsul elementelor aflate pe bordura matricii;
- b. Pe următoarele  $n+2$  linii se va afișa matricea dată, bordată cu o linie la început și una la sfârșit și cu o coloană la început și una la sfârșit având toate elementele egale cu 1.

15. Să se scrie un program care citește de la tastatură un șir de cel mult 100 de caractere litere mici și afișează în fișierul **date.out**.

- a. Pe prima linie șirul inițial modificat prin eliminarea vocalelor folosind o funcție recursivă .
- b. Pe linia a doua a fișierului șirul inițial modificat prin adăugarea la începutul și sfârșitul său a literei „A”.

16. Se citește de la tastatură un vector cu  $n$  elemente numere naturale. Sa se calculeze maximul dintre elementele vectorului.

- a. Se va folosi o funcție recursivă pentru citirea vectorului.
- b. Se va folosi o funcție recursivă pentru determinarea elementului maxim.

17. Se citește de la tastatură un număr natural  $n$ . Scrieți un program care să folosească o funcție recursivă ce primește ca parametru numărul natural  $n$  și afișează:

```
1 2 3 4 ... n
...
1 2 3
1 2
1
```

18. Se citește de la tastatură un vector cu  $n$  elemente numere naturale. Scrieți un program care să afișeze elementele vectorului care au proprietatea că sunt numere factoriale.

Se vor folosi funcții recursive pentru prelucrările necesare.

Ex: Pentru  $n=6$  și vectorul 6 7 24 100 120 46 se vor afișa numerele 6 24 120

19. Se citesc de la tastatură două șiruri de maxim 100 de caractere.

- a. Să se verifice dacă cele două șiruri sunt anagrame (conțin aceleași caractere în altă ordine) prin afișarea pe ecran a unui mesaj corespunzător, folosind o funcție recursivă care realizează această verificare.
- b. Să se șteargă din ambele șiruri toate vocalele și să se afișeze șirurile modificate.

20. Se citește din fișierul text **date.in** de pe prima linie dimensiunea  $n$  a unui vector de numere reale, iar de pe linia următoare elementele tabloului. În fișierul **date.out** se vor afișa

- a. Pe prima linie elementul maxim din vector folosind metoda Divide & Impera.
- b. Pe a doua linie elementul minim din vector folosind metoda Divide & Impera.

21. Se citește un vector de numere reale din fișierul **date.in**. Scrieți un program care afișează în fișierul **date.out** vectorul sortat crescător folosind metoda interclasării.

22. Se citește un număr real  $x \geq 1$ . Se cere să se scrie un program care calculează  $\ln(x)$  cu 3 zecimale exacte folosind o funcție recursivă. Nu se va utiliza funcția logaritmică a limbajului.

23. Se citește din fișierul **date.in** un vector cu n elemente numere naturale. Să se afișeze în fișierul **date.out** :

- a. Cmmdc dintre primul si ultimul element al tabloului pe prima linie a fișierului
- b. Cmmdc dintre cele n elemente ale tabloului folosind metoda Divide & Impera pe linia a doua a fișierului.

24. La inaugurarea unui restaurant sunt prezente mai multe persoane. Clienții își aleg din meniul pus la dispoziție câte o specialitate. Dar deocamdată restaurantul a angajat un singur bucătar care pregătește mâncărurile una după alta, deci clienții nu pot fi serviți decât pe rând. Presupunând că bucătarul se apucă de lucru după ce s-au strâns toate comenzile, stabiliți în ce ordine trebuie să pregătească specialitățile, astfel încât timpul mediu de așteptare al clienților să fie minim.

**Date de intrare:** Fișierul de intrare **menu.in** conține:

N ( $N \leq 50$ ) – numărul clienților (pe prima linie)

N numere reprezentând timpul (exprimat în minute) necesar pregătirii mâncărurilor comandate (pe a doua linie)

**Date de ieșire:** Fișierul de ieșire **menu.out** va conține:

Timpul mediu de așteptare (pe prima linie)

Numerele de ordine ale persoanelor din restaurant în ordinea în care trebuie servite (pe a doua linie)

**Exemplu:**

**menu.in**

5

30 40 20 25 60

**menu.out**

86.00

3 4 1 2 5

25. O asociație caritabilă asigură consultații medicale gratuite pentru cei fără posibilități materiale. Există un singur cabinet dotat cu aparatură medicală. Din acest motiv la un moment dat un singur medic poate face consultații. Asociația apelează la n medici de diverse specialități, care își oferă benevol serviciile. Fiecare prezintă un singur interval de-a lungul aceleiași zile, în care este disponibil. Ajutați asociația să realizeze o planificare a consultațiilor în cabinet, astfel încât numărul de medici planificați să fie maxim.

**Date de intrare:** Fișierul de intrare **medici.in** conține:

N ( $N \leq 1000$ ) – numărul de medici (pe prima linie)

N linii conținând câte 2 valori întregi mai mici sau egale cu 24: ora de început și ora de sfârșit a unei consultații

**Date de ieșire:** Fișierul de ieșire **medici.out** va conține:

Numărul de medici planificați (pe prima linie)

Numerele de ordine ale medicilor planificați (pe a doua linie)

**Exemplu:**

**medici.in**

4

2 6 20 4  
3 8 23 7  
**medici.out**  
3  
1 4 3

26. Se consideră un rucsac cu care se poate transporta o greutate maximă  $G_{max}$  și mai multe obiecte de greutate  $g_1, g_2, \dots, g_n$ , la transportul cărora se obțin câștigurile  $c_1, c_2, \dots, c_n$ . Se cere să se încarce rucsacul astfel încât să se obțină un câștig maxim, știind că obiectele pot fi „tăiate”. Rezultatul va fi afișat printr-un șir de  $n$  numere reale cuprinse între 0 și 1, având semnificația procentului transportat din fiecare obiect.

**Date de intrare:** Fișierul de intrare **rucsac.in** conține:

$G_{max}$   $n$  (pe prima linie)

Două linii cu  $n$  elemente: prima conține greutatea celor  $n$  obiecte iar a doua linie conține câștigurile obținute

**Date de ieșire:** Fișierul de ieșire **rucsac.out** va conține:

O linie cu  $n$  valori reale cu semnificația cerută în enunț și pe linia următoare câștigul obținut

**Exemplu:**

**rucsac.in**

3  
2 1 3  
2 4 6

**rucsac.out**

0 1 0.66  
8

27. Se dau mulțimile  $A = \{a_1, a_2, a_3, \dots, a_m\}$  și  $B = \{b_1, b_2, b_3, \dots, b_n\}$  având elemente numere întregi nenule. Se știe că  $m$  este mai mic sau egal cu  $n$ . Se cere să se determine o submulțime a mulțimii  $B$  ( $\{x_1, x_2, x_3, \dots, x_m\}$ ) astfel încât valoarea expresiei  $E = a_1 * x_1 + a_2 * x_2 + a_3 * x_3 + \dots + a_m * x_m$  să fie minimă și să se calculeze valoarea expresiei  $E$ .

**Date de intrare:** Fișierul de intrare **expresie.in** conține:

pe prima linie  $m$

pe a doua linie **valorile**  $a_1, a_2, a_3, \dots, a_m$

pe a treia linie  $n$

pe a patra linie **valorile**  $b_1, b_2, b_3, \dots, b_n$

**Date de ieșire:** Fișierul de ieșire **expresie.out** va conține:

pe prima linie **valorile alese din mulțimea B**

pe a doua linie **valoarea expresiei E**

**Exemplu:**

**expresie.in**

3  
-3 -1 2  
4  
1 2 3 4

**expresie.out**

1 2 4  
3

28. Pe artera principala a unui oraş sunt amplasate  $n$  staţii de autobuz. Se doreşte selectarea unui număr maxim de staţii în care să se facă oprirea unui autobuz cu condiţia ca între două staţii alăturate să fie cel puţin  $x$  metri.

**Date de intrare:** Fişierul de intrare **statii.in** conţine:

pe prima linie  $n$  (numărul staţiilor)

pe a doua linie  $n-1$  numere naturale reprezentând distanţele între staţii alăturate pe a treia linie valoarea  $x$

**Date de ieşire:** Fişierul de ieşire **statii.out** va conţine:

pe prima linie o valoare reprezentând numărul maxim de staţii selectate

pe a doua linie un şir de  $n$  valori 0 şi 1 cu următoarea semnificaţie: 1 dacă staţia  $i$  a fost selectată şi 0 dacă nu a fost selectată ( $i$  ia valori de la 1 la  $n$ )

**Exemplu:**

**statii.in**

10 60

100 50 25 25 50 10 10 80 20

**statii.out**

5

1 2 4 6 9

29. Într-un zonă turistică se găsesc  $n$  obiective turistice. Un turist doreşte să plece de la unul dintre obiective, să viziteze toate obiectivele şi să revină în punctul de plecare. Cunoscând legăturile directe existente între obiective să se determine toate variantele de traseu pe care le are la dispoziţie turistul respectiv.

**Date de intrare:** Fişierul de intrare **turist.in** conţine:

pe prima linie  $n$  (numărul obiectivelor),  $x$  (obiectivul de plecare)

pe următoarele  $n$  linii pentru fiecare obiectiv ( $1, 2, \dots, n$ ) obiectivele cu care are legătură directă

**Date de ieşire:** Fişierul de intrare **turist.out** va conţine:

pe fiecare linie numerele de ordine ale obiectivelor turistice în ordinea în care apar în traseu separate prin spaţiu

**Exemplu:**

**turist.in**

6 1

2 6

1 3 5 6

2 4 5 6

3 5

2 3 4 6

1 2 3 5

**turist.out**

1 2 3 4 5 6 1

1 2 5 4 3 6 1

-----

30. Să se genereze toate modalităţile de plată a unei sume  $S$  de bani, având la dispoziţie  $n$  tipuri de monede. Din fiecare tip sunt disponibile oricâte monede.

**Date de intrare:** Fişierul de intrare **plata.in** conţine:

pe prima linie  $n$  (numărul tipurilor de monede),  $S$  (suma de bani)

pe următoarea linie  $n$  valori naturale reprezentând valorile monedelor

**Date de ieşire:** Fişierul de intrare **plata.out** va conţine:

pe fiecare linie perechi de forma (tipul monedei, numărul monedelor de acel tip)

**Exemplu:**

**plata.in**

3 30

2 3 5

**plata.out**

(1, 15)

(1, 10)

(3, 6)

(1, 5) (3, 4)

.....

31. Să se genereze toate submulțimile unei mulțimi cu  $n$  elemente numere întregi.

**Date de intrare:** Fișierul de intrare **submult.in** conține:

pe prima linie  $n$  (numărul elementelor din mulțime)

pe următoarea linie  $n$  valori întregi reprezentând elementele mulțimii

**Date de ieșire:** Fișierul de intrare **submult.out** va conține:

pe fiecare linie elementele unei submulțimi; elementele vor fi separate prin punct și virgulă iar elementele submulțimii vor fi încadrate între acolade

**Exemplu:**

**submult.in**

4

3 -9 5 -2 1

**submult.out**

{3}

{3; -9}

{3; -9; 5}

-----

32. Să se genereze toate modalitățile de descompunere a unui număr natural dat ca sumă de numere naturale nenule.

**Date de intrare:** Fișierul de intrare **desc.in** conține:

pe prima linie  $n$  (numărul natural nenul care trebuie descompus)

**Date de ieșire:** Fișierul de intrare **desc.out** va conține:

pe fiecare linie numerele naturale nenule în care este descompus  $n$ ; numerele vor fi separate prin spațiu

**Exemplu:**

**desc.in**

5

**desc.out**

1 1 1 1 1

1 1 1 2

1 1 2 1

-----

33. O fotografie alb-negru este prezentată sub forma unei matrice binare. Ea înfățișează una sau mai multe obiecte. Porțiunile corespunzătoare obiectelor în matrice au valoarea 1. Se cere să se determine dacă fotografia reprezintă unul sau mai multe obiecte.

**Date de intrare:** Fișierul de intrare **fotografie.in** conține:

pe prima linie  $m$  și  $n$  (numărul liniilor și coloanelor din matricea care codifică fotografia)

pe următoarele  $m$  linii câte  $n$  valori 0 și 1 (codificarea fotografiei)

**Date de ieșire:** Fișierul de intrare **fotografie.out** va conține:

pe prima linie mesajul „Fotografia conține un sigur obiect” sau mesajul „Fotografia conține mai multe obiecte” urmat de două puncte și de valoarea numerică ce reprezintă numărul obiectelor din fotografie

**Exemplu:**

**fotografie.in**

4 5

0 1 1 0 1

0 0 0 0 1

1 1 1 0 1

0 1 0 0 1

**fotografie.out**

Fotografia conține mai multe obiecte: 3

34. Se dă un graf neorientat cu  $n$  vârfuri și  $m$  muchii, citit prin vectorul muchiilor.
- Să se afișeze în fișierul **“graf.out”** pe prima linie numărul total de componente conexe ale grafului citit.
  - Să se afișeze în fișierul **„graf.out”** pe următoarele linii separate componentele sale conexe.
35. Se citesc numărul  $n$  de noduri ale unui graf neorientat și adiacentele dintre ele ( $n \leq 10$ ), sub formă de perechi de noduri, din fișierul **„graf.in”**. Pe prima linie se găsește  $n$ , iar pe următoarele, câte o pereche de noduri adiacente.
- Să se afișeze pe ecran nodurile de grad maxim;
  - Să se afișeze pe ecran mesajul "Graful este hamiltonian", dacă este cazul.
36. Se dă un graf orientat  $G=(X, U)$  având  $n$  vârfuri. Se citesc din fișierul text **"graf.in"**, de pe prima linie numărul  $n$  de vârfuri, iar de pe următoarele  $n$  linii, matricea de adiacență asociată grafului.
- Să se determine și să se afișeze pe ecran nodurile cu proprietatea că au numărul maxim de incidente interioare și în același timp numărul minim de incidente exterioare.
  - Pentru un nod  $x$  citit să se afișeze pe ecran cu spații între ele toate nodurile adiacente cu acesta.
37. În fișierul text **“graf.in”** sunt memorate pe prima linie  $n$  și  $m$  reprezentând numărul de noduri, respectiv numărul de muchii ale unui graf neorientat iar pe următoarele  $m$  linii perechi de numere ce reprezintă muchiile grafului.
- Să se afișeze pe ecran nodul de grad maxim
  - Să se afișeze în fișierul **“graf.out”** o parcurgere în adâncime a grafului plecând din nodul de grad maxim.
38. Se definește o muchie a unui graf neorientat ca o înregistrare cu trei câmpuri: cele două vârfuri extremități și un cost asociat muchiei. Să se creeze un graf neorientat ca un vector de  $m$  muchii,  $n$  vârfuri și un cost asociat muchiei.
- Să se afișeze în fișierul **“graf.out”** matricea de adiacență a grafului definit.
  - Să se afișeze pe ecran muchia (muchii) de cost minim.



39. Fie un graf cu  $n$  vârfuri a cărui matrice de adiacență se citește din fișierul **“graf.in”**.
- Să se determine dacă graful citit este orientat sau nu.
  - Să se verifice dacă graful dat este graf complet.
40. Fie fișierul **“graf.in”** ce conține pe linii diferite numărul  $n$  de noduri ale unui graf orientat, matricea sa de adiacență și o secvență de noduri.
- Să se verifice dacă secvența de noduri citită din fișier reprezintă un drum în graful orientat.
  - Să se verifice ce fel de drum este elementar sau neelementar.

Subiecte propuse de:

Prof. Octavian Aspru – Colegiul Național de Informatică „Matei Basarab”  
Prof. Adelina Avram – Liceul Tehnologic Oltchim  
Prof. Nicoleta Cojocaru - Colegiul Național de Informatică „Matei Basarab”  
Prof. Violeta Grecea - Colegiul Național de Informatică „Matei Basarab”  
Prof. Anamaria Hancu – Colegiul Național „Alexandru Lahovari”  
Prof. Simona Ianc – Colegiul Național „Alexandru Lahovari”

Violeta Grecea, Adelina Avram - Problemele 1-10

Octavian Aspru – Problemele 11-14

Anamaria Hancu – Problemele 15-23

Simona Ianc – Problemele 24-33

Nicoleta Cojocaru – Problemele 34-40