

1) Subprogramul **schimba** are doi parametri

- **v**, un tablou unidimensional ce conține numere naturale nenule, cu cel mult 9 cifre;
- **n**, prin care primește o valoare naturală $2 < n < 100$, reprezentând numărul de elemente din **v**.

Subprogramul modifică ordinea elementelor astfel încât, numerele din jumătatea a doua a vectorului să fie schimbate cu cele din prima jumătate. Dacă vectorul are un număr impar de elemente, atunci poziția elementului din mijlocul tabloului rămâne neschimbată.

Exemplu: dacă $n=7$ și $v=(1,45,20,26,13,50,12)$, atunci, după reordonare, $v=(13,50,12,26,1,45,20)$.

2) Subprogramul **inserareD** are doi parametri

- **v**, un tablou unidimensional ce conține numere naturale, cu cel mult 9 cifre;
- **n**, prin care primește o valoare naturală $2 < n < 100$, reprezentând numărul de elemente din **v**.

Subprogramul inserează după fiecare număr par nenul, cel mai mare divizor diferit de numărul însuși și actualizează valoarea lui **n**.

Exemplu: dacă $n=5$ și $v=(13,50,12,7,20)$, atunci, după inserare, $v=(13,50,25,12,6,7,20,10)$ și $n=8$. Subprogramul **prodMax** are doi parametri

- **v**, un tablou unidimensional ce conține numere întregi, cu cel mult 4 cifre;
- **n**, prin care primește o valoare naturală $2 < n < 100$, reprezentând numărul de elemente din **v**.

Subprogramul returnează cel mai mare produs care se poate obține înmulțind două numere din vector.

Exemplu: dacă $n=5$ și $v=(-3,5,-20,7,2)$, atunci subprogramul returnează valoarea 60.

3) Subprogramul **minMax** are doi parametri

- **v**, un tablou unidimensional ce conține numere naturale, cu cel mult 4 cifre;
- **n**, prin care primește o valoare naturală $2 < n < 100$, reprezentând numărul de elemente din **v**.

Subprogramul returnează cel mai mic număr din vector dintre cele care apar de cele mai multe ori.

Exemplu: dacă $n=7$ și $v=(3,5,20,7,5,2,20)$, atunci subprogramul returnează valoarea 5.

4) Subprogramul **ordoneaza** are doi parametri

- **v**, un tablou unidimensional ce conține numere naturale nenule, cu cel mult 9 cifre;
- **n**, prin care primește o valoare naturală $2 < n < 100$, reprezentând numărul de elemente din **v**.

Subprogramul ordonează crescător prima jumătate a tabloului și descrescător cea de-a doua jumătate a vectorului. Dacă vectorul are un număr impar de elemente, atunci poziția elementului din mijlocul tabloului rămâne neschimbată.

Exemplu: dacă $n=7$ și $v=(10,40,2,26,13,50,12)$, atunci, după reordonare, $v=(2,10,40,26,50,13,12)$.

5) Subprogramul **secvMax** are patru parametri

- **v**, un tablou unidimensional ce conține numere naturale, cu cel mult 4 cifre;
- **n**, prin care primește o valoare naturală $2 < n < 100$, reprezentând numărul de elemente din **v**;
- **k**, prin care furnizează lungimea celei mai lungi secvențe de numere egale;
- **x**, prin care furnizează cea mai mică valoare care se află într-o secvență de lungime **k**, de numere egale.

Exemplu: dacă $n=13$ și $v=(3,5,5,5,4,4,8,8,8,9,2,2,2)$, atunci $k=3$ și $x=2$.

6) Subprogramul **crescMax** are doi parametri:

- **v**, un tablou unidimensional ce conține numere naturale, cu cel mult 4 cifre;
- **n**, prin care primește o valoare naturală $2 < n < 100$, reprezentând numărul de elemente din **v**.

Subprogramul returnează lungimea celei mai lungi secvențe de numere aflate în ordine strict crescătoare.

Exemplu: dacă $n=9$ și $v=(3, 5, 4, 7, 9, 16, 6, 8, 9)$, atunci subprogramul returnează 4.

7) Subprogramul **fibon** are doi parametri:

- **n**, prin care primește o valoare naturală $2 < n < 100$;

- **v**, un tablou unidimensional prin care furnizează șirul format cu primele n numere impare din șirul lui Fibonacci.

Exemplu: dacă $n=5$, subprogramul furnizează $v=(1, 1, 3, 5, 13)$.

8) Subprogramul **prime** are doi parametri:

- **n**, prin care primește o valoare naturală $2 < n < 100$;
- **v**, un tablou unidimensional prin care furnizează șirul format cu primele n naturale prime.

Exemplu: dacă $n=5$, subprogramul furnizează $v=(2, 3, 5, 7, 11)$.

9) Subprogramul **sir** are trei parametri:

- **n**, prin care primește o valoare naturală $2 < n < 10$;
- **v**, prin care furnizează un tablou unidimensional ce conține șirul crescător în care fiecare număr $i \in [1, n]$ apare de exact i ori;
- **m**, prin care furnizează numărul de elemente din tabloul **v**.

Exemplu: dacă $n=4$, subprogramul furnizează $v=(1, 2, 2, 3, 3, 3, 4, 4, 4, 4)$ și $m=10$.

10) Subprogramul **sum** are doi parametri:

- **v**, un tablou unidimensional ce conține numere naturale, cu cel mult 4 cifre;
- **n**, prin care primește o valoare naturală $2 < n < 100$, reprezentând numărul de elemente din **v**.

Subprogramul returnează valoarea primului element din vector cu proprietatea că este egal cu suma tuturor elementelor aflate pe poziții anterioare lui. Dacă în vector nu există un astfel de număr, subprogramul returnează -1.

Exemplu: dacă $n=6$ și $v=(3, 5, 8, 2, 18, 9)$, atunci subprogramul returnează 8.

11) Subprogramul **divizorN** are doi parametri

- **v**, un tablou unidimensional ce conține numere naturale nenule, cu cel mult 9 cifre;
- **n**, prin care primește o valoare naturală $2 < n < 100$.

Subprogramul returnează cel mai mare divizor comun al celor n numere din tabloul **v**.

Exemplu: dacă $n=6$ și $v=(10, 450, 255, 650, 500, 120)$, atunci valoarea returnată este 5.

12) Subprogramul **progresie** are doi parametri

- **v**, un tablou unidimensional ce conține numere naturale nenule, cu cel mult 9 cifre;
- **n**, prin care primește o valoare naturală $2 < n < 100$.

Subprogramul verifică dacă elementele vectorului pot fi rearanjate astfel încât acestea să formeze o progresie aritmetică cu rație pozitivă. În caz afirmativ returnează rația progresiei iar altfel returnează 0.

Exemplu: dacă $n=6$ și $v=(10, 13, 7, 4, 19, 16)$, atunci valoarea returnată este 3.

13) Subprogramul **ordonare** are doi parametri

- **v**, un tablou unidimensional ce conține numere naturale nenule, cu cel mult 9 cifre;
- **n**, prin care primește o valoare naturală $2 < n < 100$.

Subprogramul rearanjează elementele vectorului **v** astfel încât toate elementele care au valori impare să se găsească înaintea tuturor elementelor care memorează valori pare. Ordinea în cadrul elementelor pare, respectiv impare este ordinea inițială.

Exemplu: dacă $n=6$ și $v=(10, 133, 75, 40, 1, 16)$ atunci, elementele sunt rearanjate în ordinea (133, 75, 1, 10, 40, 16).

14) Subprogramul **multime** are doi parametri

- **v**, un tablou unidimensional ce conține numere naturale nenule, cu cel mult 9 cifre;
- **n**, prin care primește o valoare naturală $2 < n < 100$.

Subprogramul verifică dacă elementele vectorului formează o mulțime (toate elementele sunt distincte). În caz afirmativ returnează valoarea 1 iar altfel returnează 0.

Exemplu: dacă $n=6$ și $v=(10, 13, 7, 4, 19, 16)$ atunci valoarea returnată este 1, iar dacă $v=(10, 13, 7, 13, 19, 16)$, atunci valoarea returnată este 0.

15) Subprogramul **reuniune** are 6 parametri:

- **a, b** – tablouri unidimensionale ce conțin mulțimi formate numere naturale nenule, cu cel mult 9 cifre;
- **n, m** – numere naturale $2 < n, m < 100$.

Subprogramul furnizează prin parametrul **c**, un tablou unidimensional ce reprezintă reuniunea mulțimilor memorate în tablourile a și b, iar prin parametrul **p**, numărul de elemente din tabloul c. Ordinea elementelor în tabloul c poate fi oricare.

Exemplu: dacă $n=4$, $a=(1, 6, 7, 9)$, $m=3$, $b=(2, 6, 8)$, atunci $p=6$ și $c=(1, 2, 6, 7, 8, 9)$.

16) Subprogramul **intersecție** are 6 parametri:

- **a, b** – tablouri unidimensionale ce conțin mulțimi formate numere naturale nenule, cu cel mult 9 cifre;
- **n, m** – numere naturale $2 < n, m < 100$.

Subprogramul furnizează prin parametrul **c**, un tablou unidimensional ce reprezintă intersecția mulțimilor memorate în tablourile a și b, iar prin parametrul **p**, numărul de elemente din tabloul c. Ordinea elementelor în tabloul c poate fi oricare.

Exemplu: dacă $n=4$, $a=(1, 6, 7, 9)$, $m=4$, $b=(2, 6, 8, 9)$, atunci $p=2$ și $c=(6, 9)$.

17) Subprogramul **primUltim** are patru parametri:

- **v**, un tablou unidimensional ce conține numere naturale nenule, cu cel mult 9 cifre;
- **n**, prin care primește o valoare naturală $2 < n < 100$.
- **p** și **u**, prin care furnizează poziția primului, respectiv ultimului număr par.

Dacă în vector nu sunt cel puțin două numere pare atunci prin parametrii **p** și **u** se furnizează valoarea -1.

Exemplu: dacă $n=6$ și $v=(1, 130, 7, 4, 19, 11)$, atunci $p=1$ și $u=3$.

18) Subprogramul **schimba** are doi parametri

- **v**, un tablou unidimensional ce conține numere naturale nenule, cu cel mult 9 cifre;
- **n**, prin care primește o valoare naturală $2 < n < 100$.

Subprogramul inversează ordinea elementelor în tablou și furnizează rezultatul prin același parametru, v.

Exemplu: dacă $n=6$ și $v=(10, 13, 7, 4, 19, 16)$, atunci rezultatul furnizat este $v=(16, 19, 4, 7, 13, 10)$.

19) Subprogramul **inserare** are doi parametri

- **v**, un tablou unidimensional ce conține numere naturale nenule, cu cel mult 9 cifre;
- **n**, prin care primește o valoare naturală $2 < n < 100$.

Subprogramul inserează după fiecare element din vector un nou element, egal cu poziția pe care o ocupă în vector și actualizează valoarea lui n.

Exemplu: dacă $n=3$ și $v=(10, 4, 9)$, atunci, valorile furnizate sunt $n=6$ și $v=(10, 0, 4, 1, 9, 2)$.

20) Subprogramul **inserareM** are doi parametri

- **v**, un tablou unidimensional ce conține numere naturale nenule, cu cel mult 9 cifre;
- **n**, prin care primește o valoare naturală $2 < n < 100$.

Subprogramul inserează între oricare două elemente aflate pe poziții consecutive media aritmetică a acestora și actualizează valoarea lui n.

Exemplu: dacă $n=3$ și $v=(10, 4, 9)$, atunci, valorile furnizate sunt $n=5$ și $v=(10, 7, 4, 6.5, 9)$.

21) Subprogramul *stergeM* are doi parametri

- **v**, un tablou unidimensional ce conține numere naturale nenule, cu cel mult 9 cifre;
- **n**, prin care primește o valoare naturală $2 < n < 100$.

Subprogramul șterge din vector toate elementele ce memorează valori mai mari sau cel mult egale cu valoarea memorată în primul element și actualizează valoarea lui n .

Exemplu: dacă $n=5$ și $v=(10, 4, 19, 3, 1)$, atunci, valorile furnizate sunt $n=3$ și $v=(4, 3, 1)$.

22) Subprogramul *verifica* are doi parametri

- **v**, un tablou unidimensional ce conține numere naturale nenule, cu cel mult 9 cifre;
- **n**, prin care primește o valoare naturală $2 < n < 100$.

Subprogramul returnează valoarea 1 dacă șirul format cu elementele din vector, luate în ordine, formează un șir strict crescător sau strict descrescător și 0 altfel.

Exemplu: dacă $n=6$ și $v=(3, 5, 9, 16, 46, 90)$, atunci valoarea returnată este 1.

23) O rețea de comparatori de tip n ($n \geq 2$) și de dimensiune m ($m \geq 1$) este o secvență $(c_1, c_2, \dots, c_m)$ în care fiecare element c_i , numit comparator, este o pereche de numere întregi (j, k) cu proprietatea $1 \leq j < k \leq n$. Exemplu: Rețeaua $R = ((1, 2), (2, 3))$ are tipul 3 și dimensiunea 2. Dacă **a este un vector de n numere întregi și **R** este o rețea, notăm cu $R(a)$ vectorul obținut aplicând următoarele transformări lui **a**: pentru fiecare comparator $c_i = (j, k)$, $1 \leq i \leq m$ din **R**, în ordinea în care aceștia apar în rețea, dacă $a[j] > a[k]$ atunci în vectorul **a** interschimbăm valorile de la pozițiile j și k .**

Exemplu: Pentru $R = ((1, 2), (2, 3))$ și $a = (30, 20, 10)$ avem $R(a) = (20, 10, 30)$.

a) Fie $R = ((2, 4), (1, 2), (3, 4), (2, 3))$ și $a = (40, 30, 20, 10)$. Calculați $R(a)$ și scrieți valorile intermediare ale vectorului **a** corespunzătoare transformărilor efectuate.

b) Dați exemplu de o rețea **R** de tip 4 cu proprietatea că pentru orice vector **a** format din 4 numere întregi distincte, $R(a)$ va avea elementele ordonate crescător. Justificare.

c) Scrieți o funcție C/C++ care primește ca parametri numerele naturale n și m , o matrice **R** cu m linii și 2 coloane, reprezentând o rețea de comparatori, și un vector **a** de n numere întregi. Funcția va calcula vectorul $R(a)$ și va returna valoarea 1 dacă acesta are elementele ordonate crescător. În caz contrar, va returna valoarea 0.