

# Structuri de date

## Tema 1

1. Numim perechea  $(\mathbf{x}, \mathbf{y})$  pereche ordonată dacă  $\mathbf{x} < \mathbf{y}$ .
  - (a) Să se verifice dacă un vector conține doar perechi ordonate (care satisfac condiția de mai sus) (0.5p).
  - (b) Să se verifice dacă oricare  $\mathbf{x}$  din prima jumătate a vectorului formează o pereche ordonată cu oricare  $\mathbf{y}$  din cea de-a doua jumătate.(1p)
2. Să se verifice și să se afișeze dacă un vector este superior sau inferior. Un vector este superior dacă acesta conține mai multe elemente cu valoarea mai mare decât media aritmetică a întregului vector și inferior altfel.(1p)
3. Se consideră un vector cu  $nr$  elemente numere naturale din intervalul  $[30, 80]$ , Să se determine și să se afișeze elementul care apare de cele mai multe ori.(1p)
4. Se consideră un vector  $V$  cu  $nr$  elemente numere naturale. Să se construiască un al doilea vector  $W$ , care conține fiecare element din  $V$ , care are în componență doar cifre impare, dublicându-le.(1p)  
**Exemplu:**  $V = [1, 2, 5, 13, 27, 3, 57]$   $\rightarrow W = [1, 1, 5, 5, 13, 13, 3, 3, 57, 57]$
5. Un număr de bază  $b$  este reținut într-un **std::pair<int, int>** ce conține valoarea și baza de numerație în care acesta este reprezentat. Se citesc mai multe astfel de numere reprezentate în baze diferite. Să se determine perechile de valori egale.(1.5p)
6. Se consideră un vector de  $n$  puncte. Fiecare punct este un element de tipul unei structuri cu două câmpuri, reprezentând coordonatele spațiale  $(x, y)$ . Să se afișeze perechile de 4 puncte care pot forma un dreptunghi. Dacă nu există astfel de pereche se va afișa un mesaj corespunzător(2p)
7. Se citesc dintr-un fișier un număr de elevi. Fiecare elev are un nume, un prenume și 3 note, numere naturale. Se va folosi pentru un elev un **tuple** cu câmpurile nume și prenume de tip **string** (căutați pe net documentație) și cu

trei câmpuri de note de tip **int**. Elevii vor fi memorați într-un obiect de tip **std::vector<std::tuple<std::string, std::string, int, int, int> >**. Să se sorteze vectorul de elevi descrescător după medie și să se afișeze frumos, punând în evidență elevii cu note mai mici decât 5. (2p)

8. Se consideră o structură *fracție*, cu câmpurile de tip *int* *numarator* și *numitor*. Această structură dispune de o metodă *reductie*, care reduce fracția (ex: 12/30 devine 2/5), de funcții pentru operații aritmetice, de funcții de comparare. De asemenea o funcție de transformare în număr zecimal (ex: 2/5 = 0.4). Să se citească un vector de fracții. Fiecare dintre acestea să se reducă. Să se sorteze vectorul cât mai eficient și să se calculeze suma elementelor sale. Scrieți funcții de citire/ afișare pentru fracții. Punctajul maxim se acordă pentru rezolvarea completă, elegantă și eficientă. (3p)
9. Se consideră o matrice *matr* cu *nrows* linii și *ncols* coloane, cu *ncols* < 10, ale cărei elemente sunt numere naturale formate dintr-o singură cifră. Se consideră că fiecare coloană *col* reprezintă un număr în baza *col + 2*. Să se scrie o funcție care plasează numerele transformate în baza zece într-un vector *numbers* și returnează **true**, dacă matricea a fost validă și **false** altfel. În cazul în care matricea a fost validă să se afișeze acest vector de numere. Matricea este validă dacă toate elementele de pe coloana *col* sunt numere naturale din intervalul  $[0, col + 1)$ . (2p)

**Exemplu:** Se consideră matricea

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 0 \\ 0 & 0 & 3 \\ 1 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Atunci *numbers* = {26, 141, 305}.

10. Se consideră un teren dreptunghiular reprezentat de o matrice de dimensiuni *height* × *width*. Pe acest teren pasc văcuțe și se plimbă găște. Fiecare poziție de pe teren este marcată cu 0 sau 2. Dacă este marcată cu 2, înseamnă că acolo se află două picoare. Văcuțele sunt reprezentate de două perechi picioare aflate pe poziții alăturate (pe verticală sau pe orizontală). Găștele sunt reprezentate de poziții marcate cu 2, înconjurate de 0. Să se afișeze câte văcuțe și câte găște sunt pe teren. (1p)
11. Un profesor a studiat structura relațiilor dintre elevii săi. Pentru a reprezenta această structură, profesorul a numerotat elevii de la 1 la *n* și a construit o matrice pătratică cu *n* linii astfel:  $a(i, j) = 1$  dacă elevul *i* îl agreează pe elevul *j* și 0 altfel. Se consideră că fiecare elev se agreează pe sine însuși. (1p)

- a. Afișați pe ecran elevul (elevii dacă sunt mai mulți) care are (au) cei mai mulți prieteni și câți prieteni are (au). Se consideră prieteni doi elevi care se agreează reciproc.
- b. Există vreun elev care nu are niciun prieten?
- c. Afișați elevii care nu sunt agreeați de nimeni

**Exemplu:** se consideră 6 elevi și matricea de prietenie următoare

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

- a. Elevul 2 are cei mai mulți prieteni (2) și anume pe elevul 1 și pe elevul 5.
  - b. Elevul 3 nu are niciun prieten. El agreează elevii 4 și 5, dar este agreeat de elevii 1 și 2.
  - c. Elevul care nu este agreeat de nimeni este elevul 6.
12. Alex are un joc cu blocuri din lemn de forma unor paralelelipede dreptunghice cu baza pătrată de dimensiune fixă. Blocurile au înălțimi diferite. Alex aranjează aceste blocuri în pătrate concentrice.
- a) Să se verifice dacă un astfel de aranjament, reprezentat printr-o matrice, are formă piramidală, adică: toate blocurile dintr-un pătrat sunt mai mici decât cele aflate în regiunea din interiorul acestui pătrat (1p).
  - b) Dacă pe baza fiecărui pătrat este gravat un număr pozitiv (ce reprezintă înălțimea blocului în cm), să se scrie o funcție care calculează pentru un pătrat  $k$  înălțimea medie a cuburilor de pe acel pătrat. (1p).

**Exemplu:** a) În figura de mai jos sunt prezentate două aranjamente de blocuri. Prima construcție este piramidală. A doua nu. Se observă că blocurile marcate cu gri cu numerele 2, 3 și 1, aflate pe al doilea pătrat sunt mai mari decât unele aflate pe pătratul exterior lor, iar blocul din centru marcat cu 5 este mai mic decât blocul 6 aflat pe un pătrat exterior .

Construcție piramidală

1	2	1	0.5	1
2	3	4	5	1
1	4	11	6	0.5
1	3	5	4	2
0.5	0.5	1	2	2

Construcție care nu e piramidală

1	2	1	0.5	1
2	2	4	1	1
1	4	5	6	0.5
3	3	5	4	2
1	0.5	1	2	2

b) În exemplul din figura următoare este modelul piramidal de mai sus, în care s-au pus în evidență pătratul 1 și pătratul 2.

Pătratul nr 1

1	2	1	0.5	1
2	3	4	5	1
1	4	11	6	0.5
1	3	5	4	2
0.5	0.5	1	2	2

Pătratul nr 2

1	2	1	0.5	1
2	3	4	5	1
1	4	11	6	0.5
1	3	5	4	2
0.5	0.5	1	2	2

- înălțimea medie a blocurilor din pătratul 1 este  $(1+2+1+0.5+1+1+0.5+2+2+2+1+0.5+0.5+1+1+2)/16 = 19/16 = 0.1875$  cm
- înălțimea medie a blocurilor din pătratul 2 este  $(3+4+5+6+4+5+3+4)/8 = 34/8 = 4.25$  cm.

### Condiții necesare pentru acceptarea temei:

1. Este obligatorie utilizarea alocării dinamice a memoriei pentru vectori și matrice, precum și a containerului *vector* pentru rezolvarea a cel puțin una dintre probleme. Folosiți sintaxa C++.
2. Punctaj maxim vor primi doar rezolvările eficiente.
3. NU sunt permise variabile globale.
4. SE CERE separarea citirii datelor de rezolvarea problemei! Citiți matricile din fișier!
5. Menționați la începutul programului (în comentarii) complexitatea algoritmului (fără citire afișarea rezultatelor).

**Evaluare:** Rezolvați la alegere probleme. Fiecare problemă are alături punctajul aferent. Se acordă pentru această temă suma punctajelor problemelor rezolvate, dar maxim nota 10. Un punct este din oficiu. Se ține cont de criteriile generale de evaluare, prezentate în lista de criterii de pe platformă de e-learning de la prima unitate de învățare.