

Lucrări practice la cursul
Structuri de Date și Algoritmi
Lucrare practică nr. 2

Tema:

Probleme de parcurgere a tablourilor unidimensionale/string-uri și bidimensionale.

Scopul lucrării:

Scopul lucrării este de a familiariza studenții la practica cu cicluri de parcurgere a tablourilor unidimensionale și bidimensionale utilizând limbajul C.

Exerciții:

Varianta 1

- a) Identifică elementul cel mai mic și cel mai mare.

Input (intrare)	Output (rezultat)
$v1[] = \{3, 5, 8, 6, 2, 7, 10, 11\}$	min=2 max=11

- b) Identifică cele mai mari trei numere ale unui vector.

Input (intrare)	Output (rezultat)
$v1[] = \{3, 4, 5, 8, 9, 1, 2, 7, 10, 0\}$	num1=10 num2=9 num3=8

- c) Identifică toate elementele fără repetiții și indică numărul dublurilor. Afișează rezultatul în ordine crescătoare a dublurii.

Input (intrare)	Output (rezultat)
$v1[] = \{45, 30, 45, 35, 67, 35, 30, 89, 90, 35\}$	35=3 45=2 30=2 67=1 89=1 90=1

- d) Determină reuniunea a doi vectori, într-un alt vector.

Input (intrare)	Output (rezultat)
$v1[] = \{2, 3, 1, 7\}$ $v2[] = \{3, 5, 8, 2, 4, 6\}$	$v3[] = \{2, 3, 1, 7, 5, 8, 4, 6\}$

- e) Afișează elementele unei matrici care sunt poziționate pe diagonala principală.

Input (intrare)	Output (rezultat)
$m1[][] = \{\{1,2,3\}, \{4,5,6\}, \{7,8,9\}\}$	[1]: 1 5 9

Varianta 2

- a) Calculează suma și media elementelor.

Input (intrare)	Output (rezultat)
$v1[] = \{5,2,7,6,5,3\}$	sum=28 avg=4.67

- b) Identifică cele mai mici trei numere impare ale unui vector.

Input (intrare)	Output (rezultat)
$v1[] = \{3,5,1,8,9,2,7,10,11,0\}$	minodd1=1 minodd2=3 minodd3=5

- c) Află care două elemente adunate ale unui vector formează numărul introdus de la tastatură. Afișează rezultatul fără permutații.

Input (intrare)	Output (rezultat)
$v1[] = \{4,2,3,5,1,3,8\}$ keyboard_num=6	nums (4,2) at [1,2] nums (3,3) at [3,6] nums (5,1) at [4,5]

- d) Determină intersecția a doi vectori, într-un alt vector.

Input (intrare)	Output (rezultat)
$v1[] = \{2,3,1,7\}$ $v2[] = \{3,5,8,2,4,6\}$	$v3[] = \{2,3\}$

- e) Afișează elementele unei matrici ce sunt poziționate deasupra diagonalei principale.

Input (intrare)	Output (rezultat)
$m1[][] = \{\{1,2,3\}, \{4,5,6\}, \{7,8,9\}\}$	[2]: 2 3 6

Varianta 3

- a) Caută prezența unui număr introdus de la tastatura și află de câte ori acesta se repetă într-un vector.

Input (intrare)	Output (rezultat)
$v1[] = \{1, 3, 4, 3, 8\}$ keyboard_num=3	num 3 at v[2] num 3 at v[4]

- b) Identifică cele mai mari trei numere pare ale unui vector.

Input (intrare)	Output (rezultat)
$v1[] = \{3, 4, 5, 8, 9, 1, 2, 7, 10, 0\}$	maxeven1=10 maxeven2=8 maxeven3=4

- c) Află care două numere adunate ale unui vector sunt mai aproape de zero. Afișează rezultatul fără permutații.

Input (intrare)	Output (rezultat)
$v1[] = \{8, 3, 1, 9, -4, 7, 10, 3, 9\}$	nums (3, -4) ~ 0

- d) Determină diferența dintre doi vectori, într-un alt vector.

Input (intrare)	Output (rezultat)
$v1[] = \{2, 3, 1, 7\}$ $v2[] = \{3, 5, 8, 2, 4, 6\}$	$v3[] = \{1, 7\}$

- e) Afișează elementele unei matrici ce sunt poziționate sub diagonala principală.

Input (intrare)	Output (rezultat)
$m1[][] = \{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}\}$	[3]: 4 7 8

Varianta 4

- a) Calculează numărul și suma elementelor pozitive și negative.

Input (intrare)	Output (rezultat)
$v1[] = \{1, -2, 4, -6, 7, -1, 3\}$	numpos=4 & sumpos=15 numneg=3 & sumneg=-9

- b) Identifică cel mai mare număr impar și cel mai mare număr par ale unui vector cu condiția că $impar < par$.

Input (intrare)	Output (rezultat)
$v1[] = \{3, 4, 5, 8, 9, 1, 2, 7, 10, 0\}$	odd=9 even=10

- c) Șterge un număr dintr-un vector.

Input (intrare)	Output (rezultat)
$v1[] = \{35, 30, 89, 76, 155, 123\}$ num=76	$v1[] = \{35, 30, 89, 155, 123\}$

- d) Află care-i cel mai mare subșir crescător al unui vector și lungimea lui.

Input (intrare)	Output (rezultat)
$v1[] = \{3, 1, 4, 5, 6, 7, 1, 2, 3\}$	Max ascending substring has 4 elements: 4 5 6 7

- e) Afișează elementele unei matrici ce sunt poziționate pe diagonala secundară.

Input (intrare)	Output (rezultat)
$m1[][] = \{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}\}$	[4]: 3 5 7

Varianta 5

- a) Afișează separat numerele pare de cele impare.

Input (intrare)	Output (rezultat)
$v1[] = \{1, 2, 3, 4, 5\}$	elem odd even [1] 1 [2] 2 [3] 3 [4] 4 [5] 5

- b) Calculează suma elementelor unui vector cuprinse în diapazonul $-2+4$ fără ca vectorul să fie sortat.

Input (intrare)	Output (rezultat)
$v1[] = \{2, -1, 6, 4, -3, 7, -2, 0, 9, 2\}$	sumrange=5

- c) Aduagă un număr introdus de la tastatură într-un vector pe o locație indicată de la tastatură.

Input (intrare)	Output (rezultat)
$v1[] = \{54, 2, 6, 7, 8, 65, 21, 91\}$ num=23 loc=4	$v1[] = \{54, 2, 6, 7, 23, 8, 65, 21, 91\}$

- d) Calculează media celui mai mic subșir crescător al unui vector.

Input (intrare)	Output (rezultat)
$v1[] = \{3, 1, 4, 5, 6, 7, 1, 2, 3\}$	1 2 3 avg=2.00

- e) Afișează elementele unei matrici ce sunt poziționate deasupra diagonalei secundare.

Input (intrare)	Output (rezultat)
$m1[][] = \{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}\}$	[5]: 1 2 4

Varianta 6

- a) Afișează un vector în ordine inversă.

Input (intrare)	Output (rezultat)
$v1[] = \{1, 2, 3, 4, 5, 6, 7\}$	7 6 5 4 3 2 1

- b) Află care-i cea mai mare diferență dintre două elemente ale unui vector.

Input (intrare)	Output (rezultat)
$v1[] = \{35, 78, 30, 200, 42\}$	maxdiff=170

- c) Șterge dublurile elementelor unui vector.

Input (intrare)	Output (rezultat)
$v1[] = \{2, 3, 1, 2, 4, 3, 1, 2, 1, 3, 1, 3, 4\}$	$v1[] = \{2, 3, 1, 4\}$

- d) Intercalează un număr într-un vector de numere, fără ca să-i mărești dimensiunea.

Input (intrare)	Output (rezultat)
$v1[] = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ num=321	$v2[] = \{\}$

- e) Afișează elementele unei matrici ce sunt poziționate sub diagonala secundară.

Input (intrare)	Output (rezultat)
$m1[][] = \{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}\}$	[5]: 6 8 9

Varianta 7

- a) Afișează primele 3 elemente ale unui vector în ordine inversă, iar restul neschimbate.

Input (intrare)	Output (rezultat)
$v1[] = \{1, 2, 3, 4, 5, 6, 7\}$	3 2 1 4 5 6 7

- b) Identifică elementul care se repetă cel mai des.

Input (intrare)	Output (rezultat)
$v1[] = \{1, 2, 3, 2, 5, 2, 5\}$	num 2 is repeated 3 times

- c) Șterge elementele unui vector care se repetă o singură dată.

Input (intrare)	Output (rezultat)
$v1[] = \{2, 3, 1, 3, 1, 1, 3, 1, 3, 4\}$	$v1[] = \{3, 1, 3, 1, 1, 3, 1, 3\}$

- d) Interclasează doi vectori într-un vector de numere, fără ca să-i mărești dimensiunea, cu condiția ca înscrierea în vectorul trei pe fiecare poziție să fie efectuată doar o singură dată.

<https://www.pbinfo.ro/articole/5588/interclasarea-tablourilor>

Input (intrare)	Output (rezultat)
$v1[] = \{9, 3, 9, 9, 1\}$ $v2[] = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$	$v3[] = \{\}$

- e) Afișează elementele unei matrici ce sunt poziționate sub diagonala secundară.

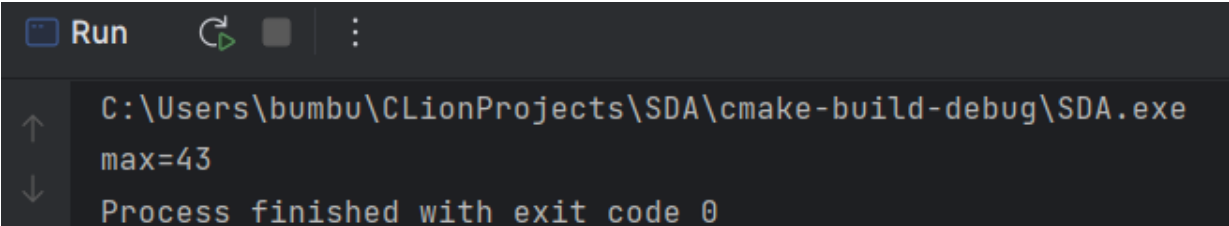
Input (intrare)	Output (rezultat)
$m1[][] = \{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}\}$	[5]: 6 8 9

Exemple de exerciții rezolvate (EX):

EX 1. Identifică elementul cel mai mare din tabloul dat.

Input (intrare)	Output (rezultat)
<code>v1[]={2,1,4,3,20,43}</code>	<code>max=43</code>

Rezolvare succintă:

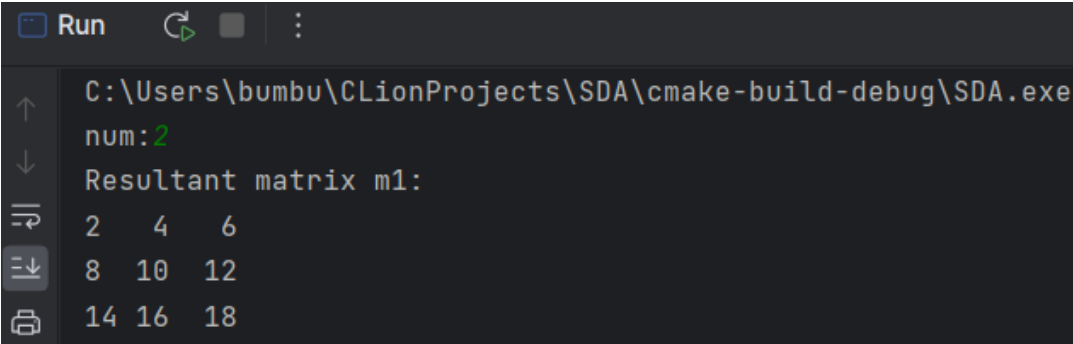
Cod sursă	Descriere cod
<pre>#include <stdio.h> int main(){ int v1[] = {2, 1, 4, 3, 43, 20}; int n = sizeof(v1) / sizeof(v1[0]); int max = v1[0]; for (int i = 1; i < n; i++){ if (v1[i] > max) max = v1[i]; } printf("max=%d", max); return 0; }</pre>	<ul style="list-style-type: none">❖ Am declarat și inițializat un tablou unidimensional 'v1'.❖ Am calculat dimensiunea tabloului 'n' prin împărțirea dimensiunii totale a tabloului la dimensiunea unui singur element (un int are 4 bytes).❖ Am declarat și inițializat variabila 'max' (adică maximum sau valoare maximă) cu primul element al tabloului 'v1'.❖ Am parcurs toate elementele tabloului cu ajutorul unui ciclu 'for'.❖ La fiecare iterație, am comparat elementul curent cu valoarea variabilei 'max' și, dacă a fost cazul, am actualizat valoarea acesteia.❖ La final, am afișat valoarea 'max' cu ajutorul funcției 'printf'.
Output:	
	

EX 2. Efectuează înmulțirea scalară a matricei cu un număr (scalar) introdus de la tastatură.

Input (intrare)	Output (rezultat)
$m1[][] = \{\{1,2,3\},\{4,5,6\},\{7,8,9\}\}$ num=2	Resultant matrix m1: 2 4 6 8 10 12 14 16 18

Rezolvare succintă:

Cod sursă	Descriere cod
<pre>#include <stdio.h> #define SIZE 3 int main(){ int m1[SIZE][SIZE] = { {1,2,3}, {4,5,6}, {7,8,9} }; int num, row, col; printf("num:"); scanf("%d", &num); /* Perform scalar multiplication of * matrix and print it in the same for */ printf("Resultant matrix m1:\n"); for(row=0; row<SIZE; row++){ for(col=0; col<SIZE; col++){ m1[row][col] *= num; if(row == 0) printf("%d ", m1[row][col]); if(row == 1) printf("%d ", m1[row][col]); if(row == 2) { if (col == 2) printf(" %d", m1[row][col]) else printf("%d ", m1[row][col]); } } printf("\n"); } return 0; }</pre>	<ul style="list-style-type: none">❖ Am declarat și inițializat un tablou unidimensional 'v1'.❖ Am calculat dimensiunea tabloului 'n' prin împărțirea dimensiunii totale a tabloului la dimensiunea unui singur element (un int are 4 bytes).❖ Am declarat și inițializat variabila 'max' (adică maximum sau valoare maximă) cu primul element al tabloului 'v1'.❖ Am parcurs toate elementele tabloului cu ajutorul unui ciclu 'for'.❖ La fiecare iterație, am comparat elementul curent cu valoarea variabilei 'max' și, dacă a fost cazul, am actualizat valoarea acestuia.❖ La final, am afișat valoarea 'max' cu ajutorul funcției 'printf'.

Output:


TODO Add more exercises

Rezultatul final

Rezultatul final al lucrării se va plasa într-un raport docx + video, unde veți înregistra procesul de elaborare a lucrării (partea cu programarea) și partea de execuție a programului. În fișierul docx veți include rezolvarea succinta a problemelor.

