**Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки**

**Лабораторна робота №3**

з дисципліни  
«Алгоритми і структури даних»

Виконав: Перевірила:

студент групи ІМ-31 Молчанова А. А.  
Литвиненко Сергій Андрійович  
номер у списку групи: 14

Київ 2023

**Завдання**

1. Написати програму розв’язання задачі пошуку (за варіантом) у двовимірному масиві (матриці) одним з алгоритмів методу лінійного пошуку.
2. Розміри матриці ***m*** та ***n*** взяти самостійно у межах від 7 до 10.
3. Виконати тестування та налагодження програми на комп’ютері. При тестуванні програми необхідно підбирати такі вхідні набори початкових значеннь матриці, щоб можна було легко відстежити коректність виконання пошуку і ця коректність була б протестована для всіх можливих випадків. З метою тестування дозволяється використовувати матриці меншого розміру.

Варіант 14:

Задано матрицю дійсних чисел A[n,n]. У головній діагоналі матриці знайти перший мінімальний і останній максимальний елементи, а також поміняти їх місцями.

**Текст програми**

#include <stdio.h>

#define SIZE 7

void printMatrix(double\* matrix, int rows, int cols) {

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

printf("%.2lf\t", \*(matrix + i \* cols + j));

}

printf("\n");

}

}

int main(int argc, char\* argv[]) {

double matrix[][SIZE] = {

{ 5.35, 3.79, -4.93, 3.31, 0.04, -8.31, -5.83 },

{ -7.33, 5.35, 6.19, -0.54, -2.31, -4.05, 2.64 },

{ -7.47, -1.39, -2.57, -2.88, 8.03, 6.88, 0.61 },

{ -8.85, -9.88, 7.63, 0.00, 2.70, 2.80, -9.33 },

{ 2.09, -4.18, 4.82, -5.82, -2.57, -6.83, 5.09 },

{ 6.14, 9.96, 7.74, 1.23, -7.13, 5.35, -1.25 },

{ -7.53, 7.17, -1.72, 2.13, 1.62, 4.55, -2.57 },

};

printf("Initial matrix:\n");

printMatrix((double\*)matrix, SIZE, SIZE);

int minPos = 0;

int maxPos = 0;

for (int i = 1; i < SIZE; i++) {

const double elem = matrix[i][i];

if (elem < matrix[minPos][minPos]) {

minPos = i;

}

else if (elem >= matrix[maxPos][maxPos]) {

maxPos = i;

}

}

printf("\nMin value = %.2lf, position = %d\n", matrix[minPos][minPos], minPos);

printf("Max value = %.2lf, position = %d\n\n", matrix[maxPos][maxPos], maxPos);

// swap

double temp = matrix[minPos][minPos];

matrix[minPos][minPos] = matrix[maxPos][maxPos];

matrix[maxPos][maxPos] = temp;

printf("Result matrix:\n");

printMatrix((double\*)matrix, SIZE, SIZE);

return 0;

}

**Тестування програми**

1. Для матриці

5.35 3.79 -4.93 3.31 0.04 -8.31 -5.83

-7.33 5.35 6.19 -0.54 -2.31 -4.05 2.64

-7.47 -1.39 -2.57 -2.88 8.03 6.88 0.61

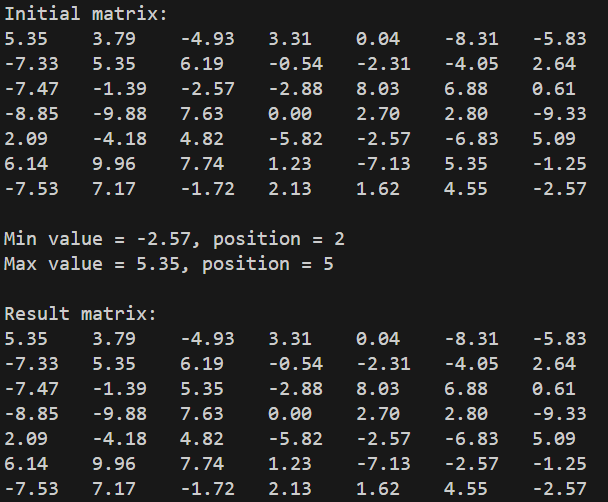
-8.85 -9.88 7.63 0.00 2.70 2.80 -9.33

2.09 -4.18 4.82 -5.82 -2.57 -6.83 5.09

6.14 9.96 7.74 1.23 -7.13 5.35 -1.25

-7.53 7.17 -1.72 2.13 1.62 4.55 -2.57

Програма видає наступний результат:



2) Для матриці

8.15 -9.75 8.19 4.27 3.67 9.87 -5.37

2.51 3.37 9.69 5.12 2.88 -7.23 9.44

-1.80 -5.16 -3.08 -7.30 -6.52 7.36 8.84

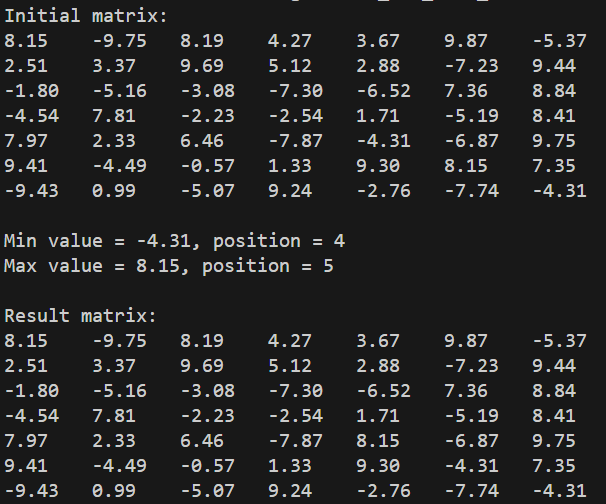
-4.54 7.81 -2.23 -2.54 1.71 -5.19 8.41

7.97 2.33 6.46 -7.87 -4.31 -6.87 9.75

9.41 -4.49 -0.57 1.33 9.30 8.15 7.35

-9.43 0.99 -5.07 9.24 -2.76 -7.74 -4.31

Програма видає наступний результат:



3) Для матриці

4 2 -10 -6 -9 7 -6

9 6 -6 -2 4 -7 -2

3 3 -2 3 1 3 5

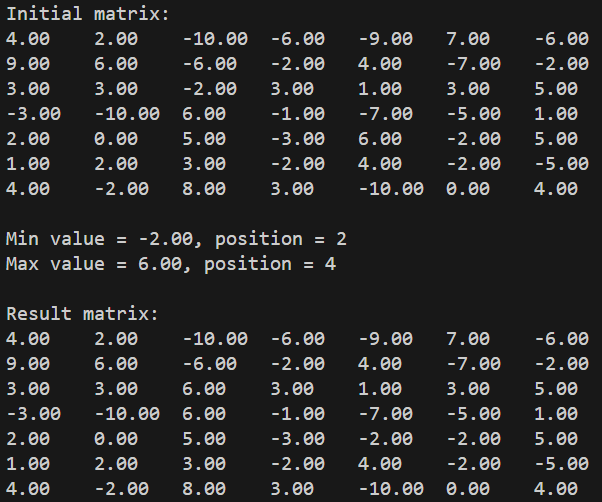
-3 -10 6 -1 -7 -5 1

2 0 5 -3 6 -2 5

1 2 3 -2 4 -2 -5

4 -2 8 3 -10 0 4

Програма видає наступний результат:



4) Для матриці

4.95 5.17 2.40 -0.25 -9.98 -3.34 7.46

0.90 4.95 -5.93 -7.16 -1.96 8.63 -8.53

-2.52 -0.70 -9.77 1.68 -2.48 3.59 3.20

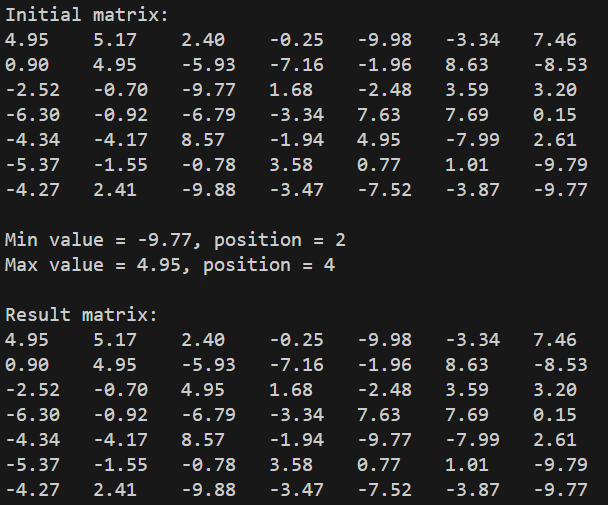
-6.30 -0.92 -6.79 -3.34 7.63 7.69 0.15

-4.34 -4.17 8.57 -1.94 4.95 -7.99 2.61

-5.37 -1.55 -0.78 3.58 0.77 1.01 -9.79

-4.27 2.41 -9.88 -3.47 -7.52 -3.87 -9.77

Програма видає наступний результат:



5) Для матриці

-6.50 6.33 -1.96 6.60 2.98 0.28 -4.30

-7.01 3.06 -9.96 -7.11 2.78 -8.99 1.68

-4.94 -0.71 -8.36 6.08 8.67 9.85 5.53

-4.90 0.99 2.15 3.06 5.80 -2.16 7.74

7.73 2.31 -1.52 -5.69 -8.36 6.96 -3.52

-1.18 -7.34 3.68 8.52 -9.03 -6.91 2.36

2.79 -0.26 -8.98 9.45 -4.03 -6.14 -8.36

Програма видає наступний результат:

