

**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 №2.1  
з дисципліни  
«Алгоритми і структури даних»**

Виконав:

Студент групи ІМ-13  
Кірович Олександр Костянтинович  
Номер у списку групи: 10

Перевірила  
Молчанова А.А

Київ 2021

### Завдання:

1. Написати програму розв'язання задачі пошуку (за варіантом) у двовимірному масиві (матриці) методом двійкового пошуку. Алгоритм двійкового пошуку задається варіантом завдання.
2. Розміри матриці  $m$  та  $n$  взяти самостійно у межах від 7 до 10.
3. При тестуванні програми необхідно підбирати такі вхідні набори початкових значень 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 № 10

Задано матрицю дійсних чисел  $A[m,n]$ . Визначити присутність серед усіх елементів матриці заданого дійсного числа  $X$  і його місцезнаходження (координати) методом двійкового пошуку (Алгоритм №2), якщо елементи кожного рядка окремо впорядковані за незменшенням.

### Текст програми:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    int rows, cols;
    printf("Number of rows: ");
    scanf("%i", &rows);
```

```

printf("Number of cols: ");
scanf("%i", &cols);
printf("Enter %i elements and X\n", rows*cols);

double matrix[rows][cols];

for (int i = 0; i < rows; i++){
    for (int j = 0; j < cols; j++)
        scanf("%lf", &matrix[i][j]);
}

double x;
scanf("%lf", &x);

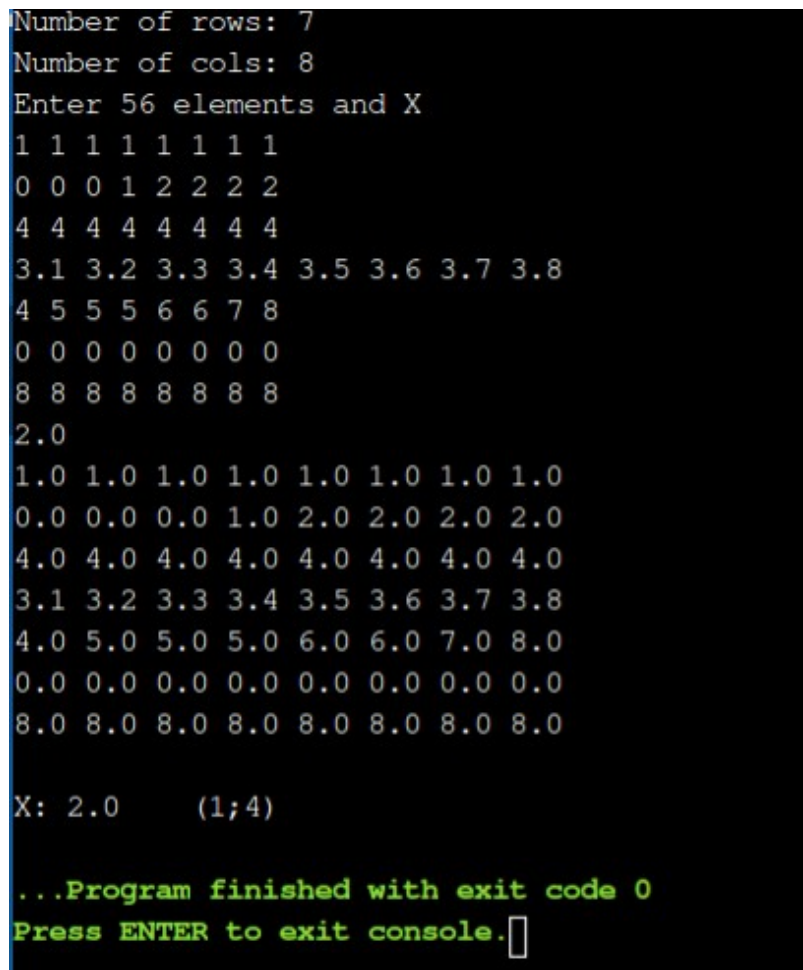
for (int i = 0; i < rows; i++){
    for (int j = 0; j < cols; j++){
        printf("%.1lf ", matrix[i][j]);
    }
    printf("\n");
}; printf("\n");

int right, pivot, left;
for (int i = 0; i < rows; i++){
    left = 0;
    right = cols;
    while (left < right){
        pivot = (left+right)/2;
        if (matrix[i][pivot] >= x){
            right = pivot;
        }
        else if (matrix[i][pivot] < x){
            left = pivot+1;
        }
    }
    if (matrix[i][pivot] == x && pivot<=cols-1){
        printf("X: %.1lf\t (%d;%d)", matrix[i][pivot], i, pivot); return 0;
    }
    else{
        pivot++;
        if (matrix[i][pivot] == x && pivot<=cols-1){
            printf("X: %.1lf\t (%d;%d)", matrix[i][pivot], i, pivot); return 0;
        }
    }
}

```

```
        if (i == cols - 1){  
            printf ("X is not there");  
        }  
    }  
    return 0;  
}
```

### Скріншоти тестування:



The screenshot shows the execution of a C program that creates a 7x8 matrix and searches for the character 'X'. The program prompts for the number of rows (7) and columns (8), then asks for 56 elements. The matrix is populated with various integers and floating-point numbers. After the input, the program displays the matrix and the search result: 'X: 2.0 (1;4)'. The program then finishes with exit code 0 and prompts the user to press ENTER to exit the console.

```
Number of rows: 7  
Number of cols: 8  
Enter 56 elements and X  
1 1 1 1 1 1 1 1  
0 0 0 1 2 2 2 2  
4 4 4 4 4 4 4 4  
3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8  
4 5 5 5 6 6 7 8  
0 0 0 0 0 0 0 0  
8 8 8 8 8 8 8 8  
2.0  
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0  
0.0 0.0 0.0 1.0 2.0 2.0 2.0 2.0  
4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0  
3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8  
4.0 5.0 5.0 5.0 6.0 6.0 7.0 8.0  
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
8.0 8.0 8.0 8.0 8.0 8.0 8.0 8.0  
  
X: 2.0      (1;4)  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```

```
Number of rows: 7
Number of cols: 7
Enter 49 elements and X
1 1 1 1 1 1 1
2 2 2 2 2 2 2
3 3 3 3 3 3 3
4 4 4 4 4 4 4
5 5 5 5 5 5 5
1 1 1 1 1 1 1
7 7 7 7 7 7 7
6.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0
2.0 2.0 2.0 2.0 2.0 2.0 2.0
3.0 3.0 3.0 3.0 3.0 3.0 3.0
4.0 4.0 4.0 4.0 4.0 4.0 4.0
5.0 5.0 5.0 5.0 5.0 5.0 5.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0
7.0 7.0 7.0 7.0 7.0 7.0 7.0

X is not there

...Program finished with exit code 0
Press ENTER to exit console. 
```

```
Number of rows: 8
Number of cols: 7
Enter 56 elements and X
8 8 8 8 8 8 8
4 5 6 7.8 9 9
0 0 0 0 0 0 0
3 3 3 4 3 3 3
1 1 1 1 1 1 1
10 10 10 10 10
11 11
0 0 0 0 0 0 0
1 1 1 1 1 1 1
4.0
4.0
8.0 8.0 8.0 8.0 8.0 8.0 8.0
4.0 5.0 6.0 7.8 9.0 9.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 3.0
3.0 3.0 4.0 3.0 3.0 3.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 10.0
10.0 10.0 10.0 10.0 11.0 11.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 4.0

X: 4.0      (1;0)

...Program finished with exit code 0
Press ENTER to exit console. 
```

```
Number of rows: 7
Number of cols: 7
Enter 49 elements and X
1 1 1 1 1 1 1
2 2 2 2 2 2 2
0 0 0 0 0 0 0
2 3 3 6 7 7 9
0 0 0 0 0 0 0
1 1 1 1 1 1 1
3 3 3 3 3 3 3
6.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0
2.0 2.0 2.0 2.0 2.0 2.0 2.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0
2.0 3.0 3.0 6.0 7.0 7.0 9.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0
3.0 3.0 3.0 3.0 3.0 3.0 3.0

X: 6.0      (3;3)

...Program finished with exit code 0
Press ENTER to exit console.□
```

```
Number of rows: 7
Number of cols: 7
Enter 49 elements and X
1 1 1 1 1 1 1
2 2 2 3 2 2 2
3 3 3 3 3 3 3
4 4 4 4 4 4 4
5 6 7 8 9 10 11
6 6 6 7 7 7 8
0 0 0 0 0 0 0
4.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0
2.0 2.0 2.0 3.0 2.0 2.0 2.0
3.0 3.0 3.0 3.0 3.0 3.0 3.0
4.0 4.0 4.0 4.0 4.0 4.0 4.0
5.0 6.0 7.0 8.0 9.0 10.0 11.0
6.0 6.0 6.0 7.0 7.0 7.0 8.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0

X: 4.0      (3;0)

...Program finished with exit code 0
Press ENTER to exit console.□
```