

**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:	Перевірила:
студент групи ІМ-21	Молчанова А. А.
Сірик Максим Олександрович	
номер у списку групи: 19	

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 19

Задано матрицю дійсних чисел $A[m,n]$. Визначити присутність серед усіх елементів матриці будь-якого з чисел діапазону $[0,5]$ і його місцезнаходження (координати) методом двійкового пошуку (Алгоритм No1), якщо елементи кожного стовпчика окремо впорядковані за незбільшенням.

Текст програми

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

```
int main (int argc, char *argv[]) {  
    printf("Lab 2.1 \n");
```

```
    float a[][3] = { {9, 9, 6}, {9, 6, 2}, {6, 6, 2}, {-1, 3, 2}, {-1, 3, 1}, {-10, 2, 0}, };
```

```
    // float a[][3] = { {9, 9, 6}, {9, 6, 2}, {9, 6, 2}, {8, 3, 2}, {8, 3, 1}, {8, 2, 0}, {5, 2,  
0}, {5, 2, 0}, {5, 2, 0}, };
```

```
    // float a[][3] = { {8, 6, 7}, {8, 6, 7}, {8, 6, 7}, {6, -1, 6}, {6, -1, 0}, {6, -1, 0}, };
```

```
    // float a[][4] = { {9, 9, 20, 6}, {9, 6, 20, 5}, {6, 6, 15, 4}, {-1, -1,  
15, 3}, {-1, -2, 10, 2}, {-10, -2, 10, 1}, };
```

```
    const int m = (int)(sizeof(a) / sizeof(a[0]));
```

```
    const int n = (int)(sizeof(a[0]) / sizeof(a[0][0]));
```

```
    printf("m = %i, n = %i \n", m, n);
```

```
printf("Printing the matrix\n");
for (int i = 0; i < m; i++) {
    for (int j = 0; j < n; j++) {
        printf("%*.1f ", 5, a[i][j]);
    }
    printf("\n");
}
```

```
for (int j = 0; j < n; j++) {
    int up = 0;
    int down = m - 1;
    while (up <= down) {
        int mid = (up + down) / 2;
        float val = a[mid][j];
        // printf("a[%i, %i] = %f\n", mid, j, val);

        if (0 <= val && val <= 5) {
            printf("Found! a[%i, %i] = %f", mid, j, val);
            return 0;
        } else if (val > 5) {
            up = mid + 1;
        } else if (val < 0) {
            down = mid - 1;
        }
    }
}
```

```
printf("There is no such element");
```

```

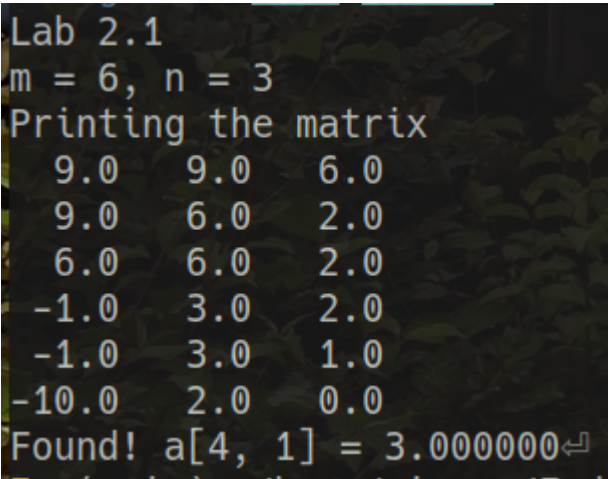
return 0;
}

```

Вхідні дані

- 1) float a[][3] = { {9, 9, 6}, {9, 6, 2}, {6, 6, 2}, {-1, 3, 2}, {-1, 3, 1}, {-10, 2, 0},
};
- 2) float a[][3] = { {9, 9, 6}, {9, 6, 2}, {9, 6, 2}, {8, 3, 2}, {8, 3, 1}, {8, 2, 0}, {5,
2, 0}, {5, 2, 0}, {5, 2, 0}, };
- 3) float a[][3] = { {8, 6, 7}, {8, 6, 7}, {8, 6, 7}, {6, -1, 6}, {6, -1, 0}, {6, -1, 0}, };
- 4) float a[][4] = { {9, 9, 20, 6}, {9, 6, 20, 5}, {6, 6, 15, 4}, {-1, -1, 15, 3}, {-1, -2,
10, 2}, {-10, -2, 10, 1}, };
- 5) float a[][4] = { {9, 9, 20, 7}, {9, 6, 20, 7}, {6, 6, 15, 7}, {-1, -1, 15, 7}, {-1, -2,
10, 7}, {-10, -2, 10, -1}, };

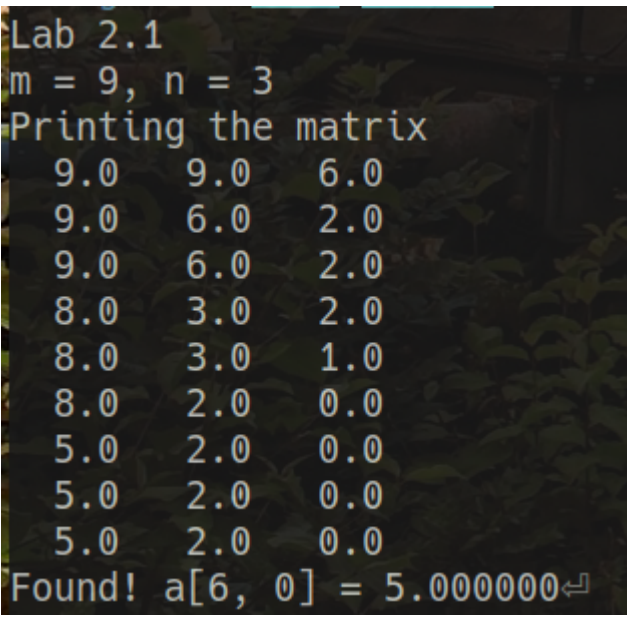
Результати тестування програми

1) 

```

Lab 2.1
m = 6, n = 3
Printing the matrix
 9.0  9.0  6.0
 9.0  6.0  2.0
 6.0  6.0  2.0
-1.0  3.0  2.0
-1.0  3.0  1.0
-10.0 2.0  0.0
Found! a[4, 1] = 3.000000

```

2) 

```

Lab 2.1
m = 9, n = 3
Printing the matrix
 9.0  9.0  6.0
 9.0  6.0  2.0
 9.0  6.0  2.0
 8.0  3.0  2.0
 8.0  3.0  1.0
 8.0  2.0  0.0
 5.0  2.0  0.0
 5.0  2.0  0.0
 5.0  2.0  0.0
Found! a[6, 0] = 5.000000

```

```
Lab 2.1
m = 6, n = 3
Printing the matrix
 8.0  6.0  7.0
 8.0  6.0  7.0
 8.0  6.0  7.0
 6.0 -1.0  6.0
 6.0 -1.0  0.0
 6.0 -1.0  0.0
3) Found! a[4, 2] = 0.000000↵
```

```
Printing the matrix
 9.0  9.0 20.0  6.0
 9.0  6.0 20.0  5.0
 6.0  6.0 15.0  4.0
-1.0 -1.0 15.0  3.0
-1.0 -2.0 10.0  2.0
-10.0 -2.0 10.0  1.0
4) Found! a[2, 3] = 4.000000↵
```

```
Lab 2.1
m = 6, n = 4
Printing the matrix
 9.0  9.0 20.0  7.0
 9.0  6.0 20.0  7.0
 6.0  6.0 15.0  7.0
-1.0 -1.0 15.0  7.0
-1.0 -2.0 10.0  7.0
-10.0 -2.0 10.0 -1.0
5) There is no such element↵
```

Для можливості перевірити виконання алгоритму, до останнього варіанту додам скрін з логуванням перебору елементів

Lab 2.1

m = 6, n = 4

Printing the matrix

9.0	9.0	20.0	7.0
9.0	6.0	20.0	7.0
6.0	6.0	15.0	7.0
-1.0	-1.0	15.0	7.0
-1.0	-2.0	10.0	7.0
-10.0	-2.0	10.0	-1.0

a[2, 0] = 6.000000

a[4, 0] = -1.000000

a[3, 0] = -1.000000

a[2, 1] = 6.000000

a[4, 1] = -2.000000

a[3, 1] = -1.000000

a[2, 2] = 15.000000

a[4, 2] = 10.000000

a[5, 2] = 10.000000

a[2, 3] = 7.000000

a[4, 3] = 7.000000

a[5, 3] = -1.000000

There is no such element↵