

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

Звіт
з лабораторної роботи №8 з дисципліни
«Основи програмування»
«Багатовимірні масиви»
Варіант 34

Виконав студент ІП-1134 Шамков Іван Дмитрович
(прізвище, ім'я, по батькові)

Перевірив викладач Вітковська Ірина Іванівна
(прізвище, ім'я, по батькові)

Київ 2021
Лабораторна робота №8
Багатовимірні масиви

Лабораторна робота 8

Багатовимірні масиви

Мета – опанувати технологія використання двовимірних масивів даних(матриць), навчитися розробляти алгоритми та програми із застосуванням матриць.

Варіант: 34

Умова задачі:

34. Для заданої матриці $B(n \times n)$ вивести усі сідлові точки та їх індекси. Сідлова точка b_{ij} визначається як мінімальне значення в i -му рядку і максимальне значення в j -му стовпці матриці. Якщо сідлових точок у цій матриці немає, то вивести відповідне повідомлення.

Математична модель:

Змінна	Тип	Ім'я	Призначення
Кількість елементів матриці	Цілий	n	Початкове дане
Двовимірний масив	Дійсний	A	Проміжне значення
Показчик на масив A	Дійсний	ptrA	Проміжне значення
Мінімальний елемент рядка, який потім перевіряємо, чи є він сідловою точкою	Дійсний	min	Результат
Номер мінімального елемента	Цілий	id	Проміжне значення
Лічильник	Цілий	i	Проміжне значення
Лічильник	Цілий	j	Проміжне значення
Лічильник	Цілий	k	Проміжне значення

Змінна для перевірки умови максимального елемента стовпчика	Логічний	Bool	Проміжне значення
---	----------	------	-------------------

Постановка задачі:

Отже, математичне формулювання нашої задачі полягає в тому, щоб створити масив двовимірний масив А із заданим розміром n, який складатиметься з дійсних чисел. У рядках масиву шукаємо мінімальний елемент. Після цього запам'ятовуємо його індекс. Це буде номер стовпчика, який нам потрібно перевірити. Проходимося тепер по цьому стовпчику та перевіряємо, чи є знайдений елемент найбільшим у ньому. Якщо так, то виводимо його. Якщо ні то продовжуємо процедуру, поки не дійдемо до кінця.

Текст:

```
#include <iostream>
```

```
#include<ctime>
```

```
#include <iomanip>
```

```
using namespace std;
```

```
float** CreateArray(int);
```

```
void FindPoint(float**, int);
```

```
void CoutArray(float**, int);
```

```
int main() {
```

```
    int n;
```

```
    srand(time(NULL));
```

```
    cout << "Enter the size of two dimensional array: ";
```

```

    cin >> n;

    cout << "\n";

    float** ptrA;

    ptrA = CreateArray(n);

    CoutArray(ptrA, n);

    FindPoint(ptrA, n);

    delete[] ptrA;

    return 0;

}

float** CreateArray(int n) {
    float** A= new float* [n];

    for (int i=0; i < n; i++) {
        A[i] = new float[n];

    }

    for (int i=0; i < n; i++) {
        for (int j=0; j < n; j++) {
            A[i][j] =float( rand()%2000-1000)/10;

        }
    }

    return A;

}

void DeleteArray(float** A) {

```

```

        delete []A;
    }

void CoutArray(float** A, int n) {
    cout << "\tTwo dimensional Array:\n\n";
    for (int i=0; i < n; i++) {
        for (int j=0; j < n; j++) {
            cout << setw(7) << A[i][j];
        }
        cout << "\n\n";
    }
    cout << "\n\n";
}

```

```

void FindPoint(float** A, int n) {
    bool Bool;
    float min;
    int id;
    for (int i = 0; i < n; i++) {
        min = A[i][0];
        id = 0;
        Bool = true;
        for (int j = 1; j < n; j++) {
            if (min > A[i][j]) {
                min = A[i][j];
                id = j;
            }
        }
    }
}

```

```

    int k = 0;

    for ( ; k < n && Bool; k++) {
        if (min <= A[k][id] && k!=i) {
            Bool = false;
        }

    }

    if (Bool)
        cout << "The point is number " << setw(4) << min << " and
it's id: [" << i << "] [" << id << "]" << endl;
    else cout << "No such points in a row " << i << endl;
    k = 0;
}
}

```

Текст файла проекту:

C++:

```
#include <iostream>
#include<ctime>
#include <iomanip>
using namespace std;
float** CreateArray(int);
void FindPoint(float**, int);
void CoutArray(float**, int);

int main() {
    int n;
    srand(time(NULL));
    cout << "Enter the size of two dimensional array: ";
    cin >> n;
    cout << "\n";
    float** ptrA;

    ptrA = CreateArray(n);
    CoutArray(ptrA, n);
    FindPoint(ptrA, n);

    delete[] ptrA;
    return 0;
}
```

```

24
25 float** CreateArray(int n) {
26     float** A= new float* [n];
27
28     for (int i=0; i < n; i++) {
29         A[i] = new float[n];
30     }
31
32     for (int i=0; i < n; i++) {
33         for (int j=0; j < n; j++) {
34             A[i][j] =float( rand()%2000-1000)/10;
35         }
36     }
37     return A;
38 }
39
40
41
42 void CoutArray(float** A, int n) {
43     cout << "\tTwo dimensional Array:\n\n";
44     for (int i=0; i < n; i++) {
45         for (int j=0; j < n; j++) {
46             cout << setw(7) << A[i][j];
47         }
48         cout << "\n\n";
49     }
50     cout << "\n\n";
51 }
52

```

```

52
53 void FindPoint(float** A, int n) {
54     bool Bool;
55     float min;
56     int id;
57     for (int i = 0; i < n; i++) {
58         min = A[i][0];
59         id = 0;
60         Bool = true;
61         for (int j = 1; j < n; j++) {
62             if (min > A[i][j]) {
63                 min = A[i][j];
64                 id = j;
65             }
66         }
67         int k = 0;
68         for ( ; k < n && Bool; k++) {
69             if (min <= A[k][id] && k!=i) {
70                 Bool = false;
71             }
72         }
73         if (Bool)
74             cout << "The point is number " << setw(4) << min << " and it's id: [" << i << "] [" << id << "]" << endl;
75         else cout << "No such points in a row " << i << endl;
76         k = 0;
77     }
78 }
79

```


Копії екранних форм:

C++:

```
Enter the size of array: 3

    Two dimensional Array:

    -9    -8    -9

    -4     3    -6

     0    -7    -8

No such points in a row 0
The point is number    -6 and it's id: [1] [2]
No such points in a row 2
```

```
Enter the size of array: 5
```

```
Two dimensional Array:
```

```
32.149 19.691 25.867 32.113 30.812
```

```
18.425 12.791 23.466 4.349 23.917
```

```
26.578 6.832 26.723 30.489 9.079
```

```
5.382 7.118 30.41 21.2 1.792
```

```
9.243 10.362 11.835 12.94 14.918
```

```
The point is number 19.691 and it's id: [0] [1]
```

```
No such points in a row 1
```

```
No such points in a row 2
```

```
No such points in a row 3
```

```
No such points in a row 4
```

 Select Microsoft Visual Studio Debug Console

```
Enter the size of array: 4
```

```
Two dimensional Array:
```

```
32.341 31.255 31.277 10.056
```

```
16.779 20.726 32.585 14.597
```

```
28.261 21.629 17.852 9.431
```

```
22.196 23.683 7.697 12.522
```

```
No such points in a row 0
```

```
The point is number 14.597 and it's id: [1] [3]
```

```
No such points in a row 2
```

```
No such points in a row 3
```

Висновок

Отже, виконавши цю лабораторну роботу, ми навчилися працювати з багатовимірними масивами. Проекти, на мою думку, розроблені коректно, адже заплановані елементи працюють, а саме: отримання даних від користувача, виклик функцій створення масива та наповнення його випадковими числами, пошуку найменшого елемента рядка та його індекса, перевірка, чи є цей елемент найбільшим у своєму стовпчику, а також виведення усіх цих значень на екран. Масив створюється автоматично через підпрограму `CreateArray(int)`, яка отримує значення довжини масива, а потім повертає покажчик на перший елемент масиву. Наповнюємо масив елементами з діапазону від -1000 до 1000, що ділимо на 10, щоб отримувати не цілі числа.