

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 № 9 з дисципліни

«Алгоритми та структури даних-1.

Основи алгоритмізації»

«ДСЛІДЖЕННЯ АЛГОРИТМІВ ОБХОДУ МАСИВІВ»

Варіант 1

Виконав студент ІІІ-15, Багачок Вадим Дмитрович

(шифр, прізвище, ім'я, по батькові)

Перевірів - Вечерковська А.С

(пріз

вище, ім'я, по батькові)

Київ 2021

Лабораторна робота 9
ДСЛІДЖЕННЯ АЛГОРИТМІВ ОБХОДУ МАСИВІВ
Варіант-1

Мета – дослідити алгоритми обходу масивів, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій.

№	Опис варіанту
1	Задано матрицю дійсних чисел $A[m,n]$. В кожному рядку матриці визначити присутність заданого дійсного числа X і його місцезнаходження. Обміняти знайдене значення X з елементом побічної діагоналі

Постановка задачі:

Потрібно розробити алгоритм та написати програму, яка складається з наступних дій:

- 1) Описати змінну індексованого типу (двовимірний масив) згідно з варіантом (табл. 1).
- 2) Ініціювати змінну, що описана в п.1 даного завдання
- 3) Обчислити змінну, що описана в п.1, згідно з варіантом

Матиматична модель :

Описуємо змінну індексованого типу (двовимірний масив) згідно з варіантом. заповнити змінну випадковими числами за допомогою циклу `for` . Щоб обчислити змінну x використовуємо цикл `for`.

Змінна	Тип	Ім'я	Призначення
$A[][]$	дійсне	<code>array[][]</code>	Проміжні дані Кінцеві дані
m	ціле	<code>columnSize</code>	Початкові дані, Проміжні дані
n	ціле	<code>lineSize</code>	Початкові дані, Проміжні дані
i	ціле	<code>i</code>	Проміжні дані
j	ціле	<code>j</code>	Проміжні дані
tmp	дійсне	<code>tmp</code>	Проміжні дані
X	дійсне	<code>findValue</code>	Проміжні дані
<code>valueInterval</code>	дійсне	<code>valueInterval</code>	Проміжні дані

Псевдокод :

Початок

```

columnSize:= random;
lineSize:= columnSize;
findValue:= random;
valueInterval:= random;
array[][]
повторити для i від 0 до columnSize
    повторити для j від 0 до lineSize
        array[i][j]:= random % valueInterval;

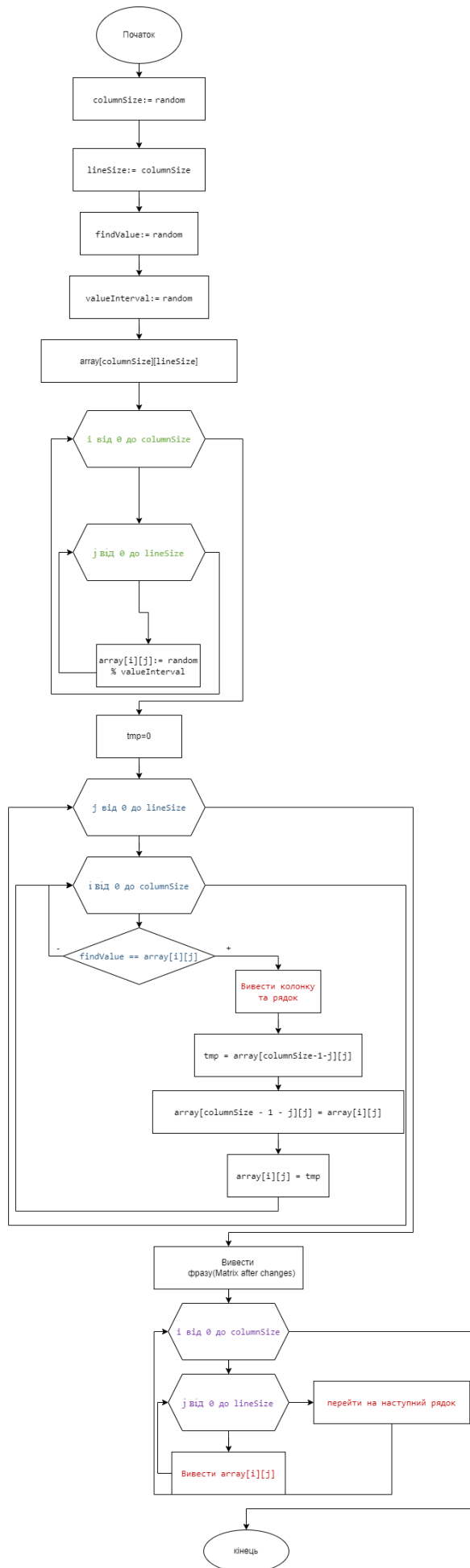
    все повторити
все повторити

повторити для j від 0 до lineSize
    повторити для i від 0 до columnSize
        якщо findValue == array[i][j]
            то
                Вивести колонку та рядок
                tmp = array[columnSize-1-j][j]
                array[columnSize - 1 - j][j] = array[i][j]
                array[i][j] = tmp

            все якщо
        все повторити
    все повторити
Вивести фразу(Matrix after changes)
    повторити для i від 0 до columnSize
        повторити для j від 0 до lineSize
            Вивести array[i][j]
        Все повторити
    перейти на наступний рядок
все повторити

```

Кінець



```

ACD Лаб-9 (Глобальная область) main()
using namespace std;
int main()
{
    string inputString;
    cout << "Enter size \n";
    getline(cin, inputString);
    int columnSize = stoi(inputString);
    int lineSize = columnSize;
    cout << "Enter the value you want to find \n";
    getline(cin, inputString);
    float findValue = stof(inputString);
    cout << "Enter the interval from which the values will be framed \n";
    getline(cin, inputString);
    int valueInterval = stoi(inputString);
    float** array = new float* [columnSize];
    for (int i = 0; i < columnSize; i++)
    {
        array[i] = new float[lineSize];
    }
    for (int i = 0; i < columnSize; i++)
    {
        for (int j = 0; j < lineSize; j++)
        {
            array[i][j] = rand() % valueInterval;
        }
    }

    for (int j = 0; j < lineSize; j++)
    {
        for (int i = 0; i < columnSize; i++)
        {
            if (findValue == array[i][j])
            {
                cout << "index - " << columnNumber - " + to_string(i) + " lineNumber - " + to_string(j) + "\n";
                float tmp = array[columnSize - 1 - j][j];
                array[columnSize - 1 - j][j] = array[i][j];
                array[i][j] = tmp;
            }
        }
    }
}

```

```

    for (int j = 0; j < lineSize; j++)
    {
        for (int i = 0; i < columnSize; i++)
        {
            if (findValue == array[i][j])
            {
                cout << "index - " << columnNumber - " + to_string(i) + " lineNumber - " + to_string(j) + "\n";
                float tmp = array[columnSize - 1 - j][j];
                array[columnSize - 1 - j][j] = array[i][j];
                array[i][j] = tmp;
            }
        }
    }

    cout << "Matrix after changes";
    for (int i = 0; i < columnSize; i++)
    {
        for (int j = 0; j < lineSize; j++)
        {
            cout << to_string(array[i][j]);
        }
        cout << "\n";
    }
}

```

```

#include <iostream>
#include<string>
#include<math.h>
using namespace std;
int main()
{
    string inputString;
    cout << "Enter size \n";
    getline(cin, inputString);
    int columnSize = stoi(inputString);
    int lineSize = columnSize;
    cout << "Enter the value you want to find \n";
    getline(cin, inputString);
    float findValue = stof(inputString);
    cout << "Enter the interval from which the values will be framed \n";
    getline(cin, inputString);
    int valueInterval = stoi(inputString);
}

```

```

float** array = new float* [columnSize];
for (int i = 0; i < columnSize; i++)
{
    array[i] = new float[lineSize];
}
for (int i = 0; i < columnSize; i++)
{
    for (int j = 0; j < lineSize; j++)
    {
        array[i][j] = rand() % valueInterval;
    }
}

for (int j = 0; j < lineSize; j++)
{
    for (int i = 0; i < columnSize; i++)
    {
        if (findValue == array[i][j])
        {
            cout << "index - " << columnNumber - " + to_string(i) + " lineNumber - "
+ to_string(j) + "\n";
            float tmp = array[columnSize-1-j][j];
            array[columnSize - 1 - j][j] = array[i][j];
            array[i][j] = tmp;
        }
    }
}
cout << "Matrix after changes";
for (int i = 0; i < columnSize; i++)
{
    for (int j = 0; j < lineSize; j++)
    {
        cout << to_string(array[i][j]);
    }
    cout << "\n";
}
}

```

The screenshot shows the 'Консоль отладки Microsoft Visual Studio' (Visual Studio Debug Console) window. It displays the following text:

```

Enter size
10
Enter the value you want to find
6
Enter the interval from which the values will be framed
10
index - columnNumber - 6 lineNumber - 0
index - columnNumber - 7 lineNumber - 0
index - columnNumber - 9 lineNumber - 0
index - columnNumber - 3 lineNumber - 1
index - columnNumber - 8 lineNumber - 1
index - columnNumber - 9 lineNumber - 2
index - columnNumber - 8 lineNumber - 4
index - columnNumber - 6 lineNumber - 5
index - columnNumber - 7 lineNumber - 6
index - columnNumber - 8 lineNumber - 6
index - columnNumber - 2 lineNumber - 7
index - columnNumber - 8 lineNumber - 7
index - columnNumber - 1 lineNumber - 9
Matrix after changes1.000007.000004.000000.000009.000004.000008.000008.000002.000006.000000
5.000005.000001.000007.000001.000001.000005.000002.000007.000004.000000
1.000004.000002.000003.000002.000002.000001.000006.000008.000005.000000
7.000003.000001.000008.000009.000002.000006.000009.000005.000004.000000
3.000001.000002.000003.000003.000006.000001.000001.000003.000008.000000
7.000004.000002.000007.000006.000006.000009.000003.000001.000009.000008.000000
4.000005.000000.000002.000008.000004.000000.000002.000004.000008.000000
6.000005.000006.000009.000000.000000.000007.000001.000003.000008.000000
9.000006.000004.000004.000007.000000.000006.000006.000001.000008.000000
6.000009.000000.000003.000007.000008.000008.000002.000009.000001.000000

```

```
Консоль отладки Microsoft Visual Studio
Enter the value you want to find
18
Enter the interval from which the values will be framed
20
index - columnNumber - 3 lineNumber - 3
index - columnNumber - 6 lineNumber - 3
index - columnNumber - 9 lineNumber - 5
index - columnNumber - 0 lineNumber - 6
index - columnNumber - 3 lineNumber - 6
index - columnNumber - 9 lineNumber - 6
index - columnNumber - 0 lineNumber - 7
index - columnNumber - 2 lineNumber - 7
index - columnNumber - 2 lineNumber - 8
index - columnNumber - 5 lineNumber - 9
Matrix after changes1.0000007.00000014.0000000.0000009.0000004.0000007.00000016.0000002.00000018.000000
5.0000005.0000001.0000007.0000001.00000011.00000015.0000002.00000018.00000016.000000
11.0000004.0000002.00000013.00000012.0000002.0000001.00000018.0000007.00000015.000000
7.0000006.00000011.0000002.0000009.00000012.00000018.00000019.00000015.00000014.000000
3.00000011.0000002.00000013.00000013.00000018.0000001.00000011.00000013.0000008.000000
7.0000004.0000002.00000017.00000017.00000019.0000003.0000001.0000009.0000004.000000
16.00000015.00000010.00000018.0000008.0000006.0000000.0000002.0000004.0000008.000000
6.0000005.00000010.0000009.00000010.00000010.0000006.0000001.00000013.0000008.000000
9.0000003.0000004.00000014.00000016.0000000.0000006.00000016.00000011.0000008.000000
4.00000019.0000006.0000003.00000017.0000004.00000018.0000002.0000009.0000001.000000
C:\Проекты КПИ\C++\АСД Лаб-9\Debug\АСД Лаб-9.exe (процесс 13556) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
2
```

Висновки: Ми дослідили алгоритми обходу масивів, набули практичних навичок використання цих алгоритмів під час складання програмних специфікацій.