

Лабораторна робота №8. Одновимірні масиви

Міністерство освіти і науки України
Національний технічний університет України «Київський
політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації
і управління

Звіт

з лабораторної роботи № 8 з дисципліни
«Основи програмування»

«Багатовимірні масиви»

Варіант __4__

Виконав студент _____ Берлінський Ярослав Владленович _____
(шифр, прізвище, ім'я, по батькові)

Перевірив _____
(прізвище, ім'я, по батькові)

Київ 2020

Назва роботи: робота з багатовимірними масивами.

Варіант: 4

Умова задачі:

4. Задана дійсна квадратна матриця розмірності n , яка містить і від'ємні елементи. Знайти найбільший за модулем її елемент. Отримати квадратну матрицю розмірності $n-1$ шляхом вилучення із вхідної матриці рядка та стовпчика, на перетині яких знаходиться знайдений елемент.

Постановка задачі. Генерується квадратна матриця дійсних чисел. Знаходиться максимальний за модулем її елемент. З матриці видаляються рядок та стовпчик, де знаходиться отриманий елемент. Виводиться нова матриця.

Розв'язок:

1. C++:

```

// Created by Берлінський Ярослав Владленович on 11.11.2020.
#include <iostream>
#include <iomanip>
#include <cmath>
#include <stdlib.h>
using namespace std;
typedef float matrix[10][10];
int n;
void matrix_generation(matrix, int, int, int);
void matrix_output(matrix, int, int);
float abs_maximum(matrix, int*, int*);
void delete_row_and_column(matrix, int*, int*);
int main(){
    int begin, end, eps;
    float max;
    matrix A;
    int row, column;
    do{
        cout<<"Введіть розмірність квадратної матриці: "; cin>>n;
        cout<<"Нижня межа генерації: "; cin>>begin;
        cout<<"Верхня межа генерації: "; cin>>end;
        cout<<"Кількість розрядів після коми: "; cin>>eps;
    }
    while(eps<0 || begin>end || n<=0);
    matrix_generation(A, begin, end, eps);
    matrix_output(A, n, eps);
    max=abs_maximum(A, &row, &column);
    cout<<"Максимальний за модулем елемент: "<<max<<endl;
    cout<<"Фізичний номер: "<<"["<<row<<"]["<<column<<"] "<<endl;
    cout<<"Логічний номер: "<<"["<<row+1<<"]["<<column+1<<"] "<<endl;
    delete_row_and_column(A, &row, &column);
    matrix_output(A, n, eps);
}
float abs_maximum(matrix A, int *prow, int *pcol){
    float max_abs=A[0][0];
    for (int i=0; i<n; i++){
        for (int j=0; j<n; j++){
            if (fabs(A[i][j])>=fabs(max_abs)){
                max_abs=A[i][j];
                *prow=i;
                *pcol=j;
            }
        }
    }
}

```

```

        *prow=i;
        *pcol=j;
    }
}
}
return max_abs;
};

void matrix_generation(matrix A, int begin, int end, int eps){
    srand(time(NULL));
    cout<<"Матрица: "<<endl;
    for (int i=0; i<n; i++)
        for (int j=0; j<n; j++)
            A[i][j]=begin+rand()%(end-begin)+(float)(rand()%((int)(pow(10,eps)))/pow(10,eps));
}

void matrix_output(matrix A, int n, int eps){
    for (int i=0; i<n; i++){
        for(int j=0; j<n; j++)
            cout<<fixed<<setprecision(eps)<<setw(4*eps)<<A[i][j];
        cout<<endl;
    }
}

void delete_row_and_column(matrix A, int *prow, int *pcol){
    for (int i=*prow; i<n; i++)
        for (int j=0; j<n; j++)
            A[i][j]=A[i+1][j];

    for (int i=0; i<n; i++)
        for (int j=*pcol; j<n; j++)
            A[i][j]=A[i][j+1];
    n--;
}

```

Тестування вхідних даних(C++):

1) n=7, begin=-100, end=100, eps=2:

```
Введіть розмірність квадратної матриці: 7
Нижня межа генерації: -100
Верхня межа генерації: 100
Кількість розрядів після коми: 2
Матриця:
  78.94   19.58  -49.83  -19.28   -7.67  -26.09  -93.16
 -54.18 -12.95   21.57 -65.64   67.24   41.49   -3.52
  91.39 -37.47   82.56 -55.58  -41.82   89.66  -33.44
  65.33   33.82 -42.04   67.72  -42.82  -10.17   41.57
 -96.55   14.30 -13.82  -44.45    0.99   39.77   40.16
  46.90   31.83   12.91   -1.35  -61.01  -91.24  -70.34
 -74.27 -58.60  -72.90   24.39   66.71  -29.21   -4.49
Максимальний за модулем елемент: -96.55
Фізичний номер: [4][0]
Логічний номер: [5][1]
  19.58  -49.83  -19.28   -7.67  -26.09  -93.16
 -12.95   21.57  -65.64   67.24   41.49   -3.52
 -37.47   82.56  -55.58  -41.82   89.66  -33.44
  33.82  -42.04   67.72  -42.82  -10.17   41.57
  31.83   12.91   -1.35  -61.01  -91.24  -70.34
 -58.60  -72.90   24.39   66.71  -29.21   -4.49
Program ended with exit code: 0|
```

2) $n=4$, $\text{begin}=-1$, $\text{end}=1$, $\text{eps}=4$:

```
Введіть розмірність квадратної матриці: 4
Нижня межа генерації: -1
Верхня межа генерації: 1
Кількість розрядів після коми: 4
Матриця:
    -0.6958      0.5628      -0.2225      -0.5217
      0.7099     -0.4353     -0.3313      0.7907
     -0.4417     -0.5341     -0.8938      0.4826
     -0.3166      0.1853      0.9430      0.7671
Максимальний за модулем елемент: 0.9430
Фізичний номер: [3][2]
Логічний номер: [4][3]
    -0.6958      0.5628      -0.5217
      0.7099     -0.4353      0.7907
     -0.4417     -0.5341      0.4826
Program ended with exit code: 0
```

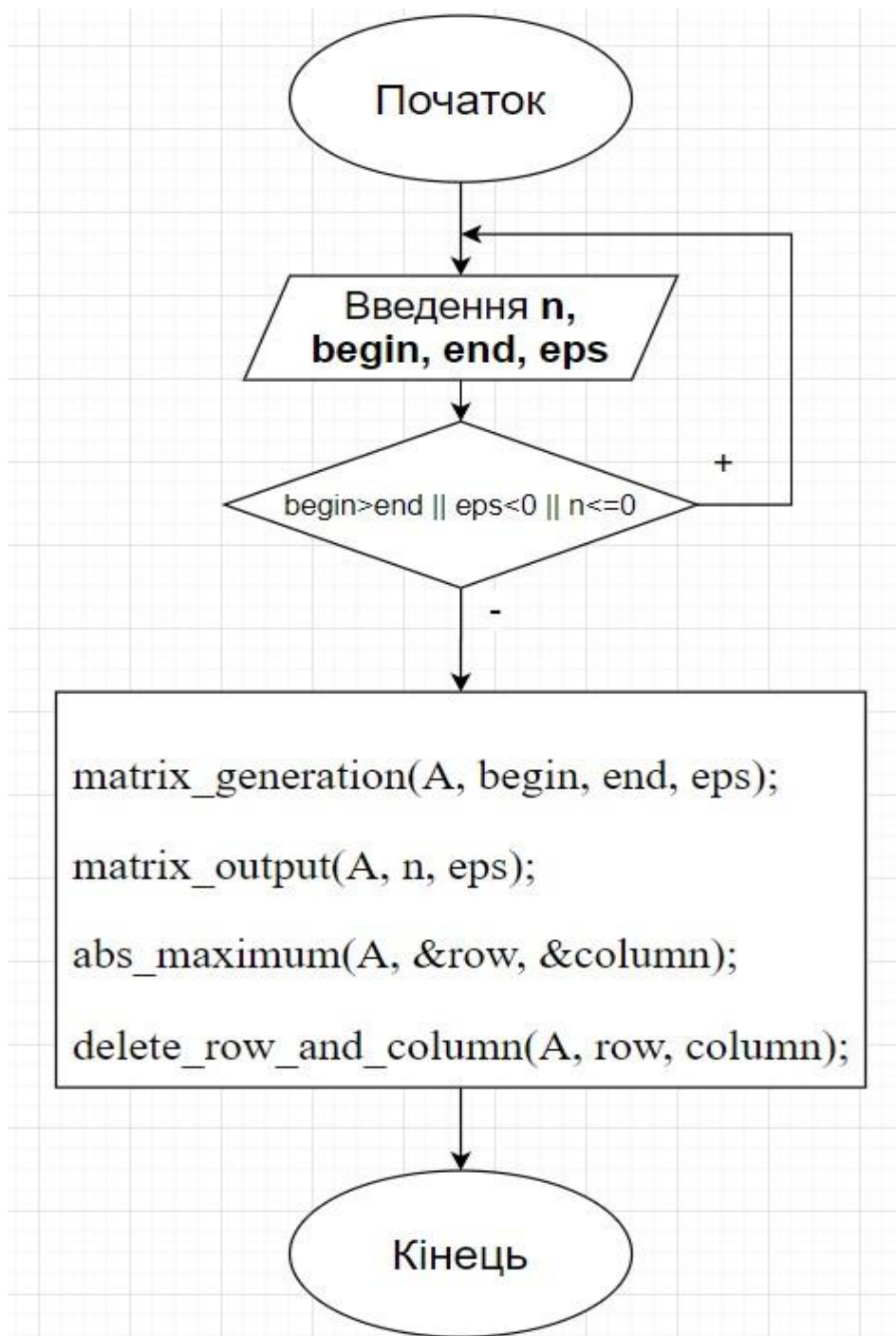
3) $n=5$, $\text{begin}=0$, $\text{end}=1$, $\text{eps}=3$:

```
Введіть розмірність квадратної матриці: 5
Нижня межа генерації: 0
Верхня межа генерації: 1
Кількість розрядів після коми: 3
Матриця:
    0.404      0.988      0.750      0.738      0.717
    0.374      0.123      0.957      0.321      0.851
    0.932      0.564      0.481      0.102      0.443
    0.011      0.176      0.890      0.004      0.962
    0.172      0.208      0.580      0.216      0.143
Максимальний за модулем елемент: 0.988
Фізичний номер: [0][1]
Логічний номер: [1][2]
    0.374      0.957      0.321      0.851
    0.932      0.481      0.102      0.443
    0.011      0.890      0.004      0.962
    0.172      0.580      0.216      0.143
Program ended with exit code: 0
```

Нескладно перевірити вірність результатів. Інші дані можна перевірити самостійно, перейшовши безпосередньо до коду програми:

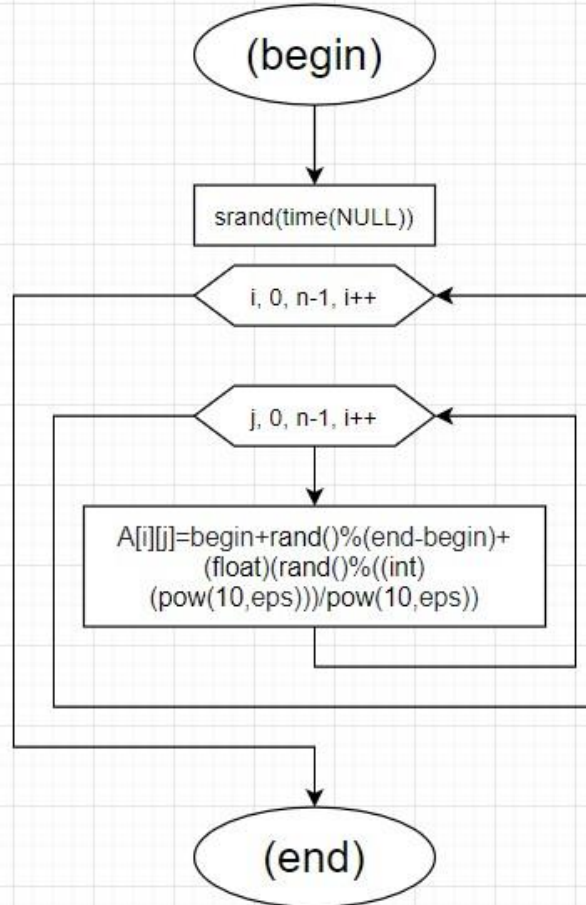
[GitHub](#)

Блок-схема роботи алгоритму:



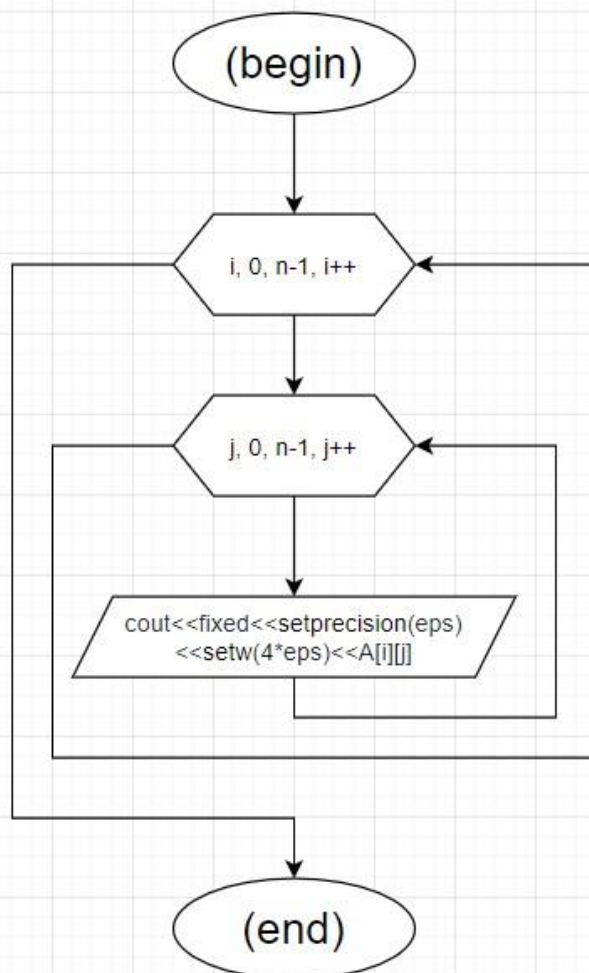
Підпрограма(1)

```
void matrix_generation(matrix A, int begin,  
int end, int eps)
```



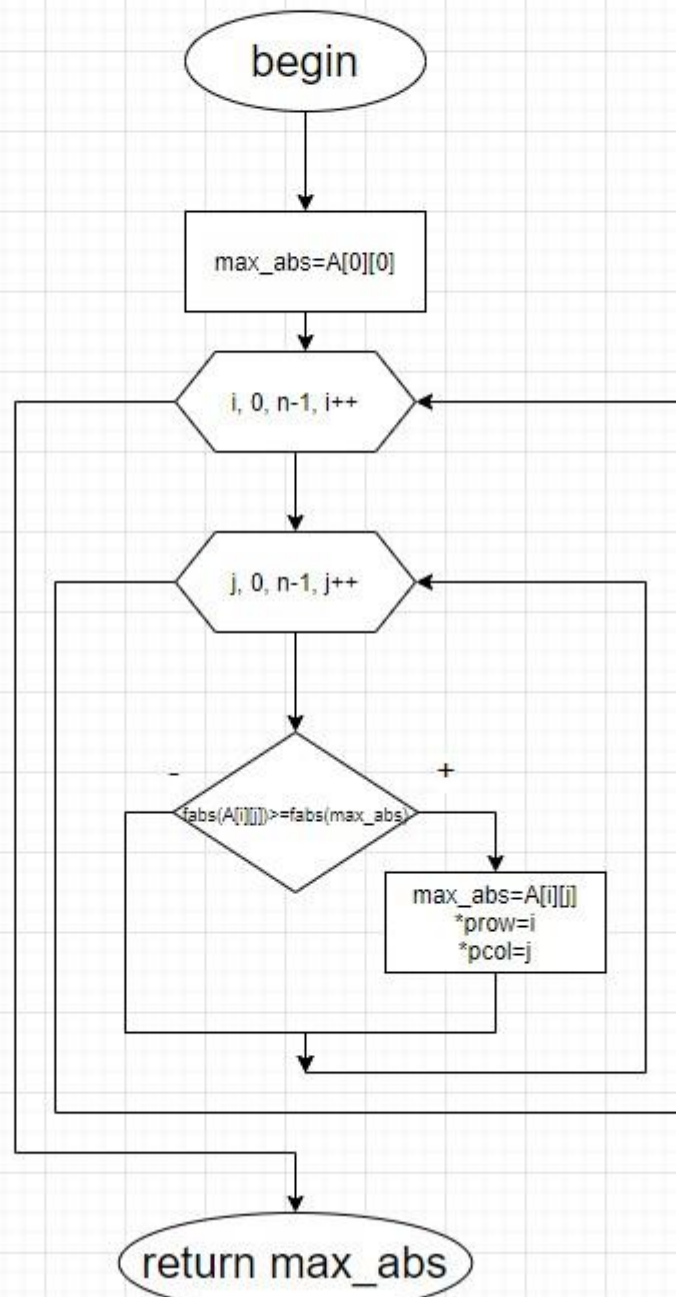
Підпрограма(2)

void matrix_output(matrix A, int n, int eps)



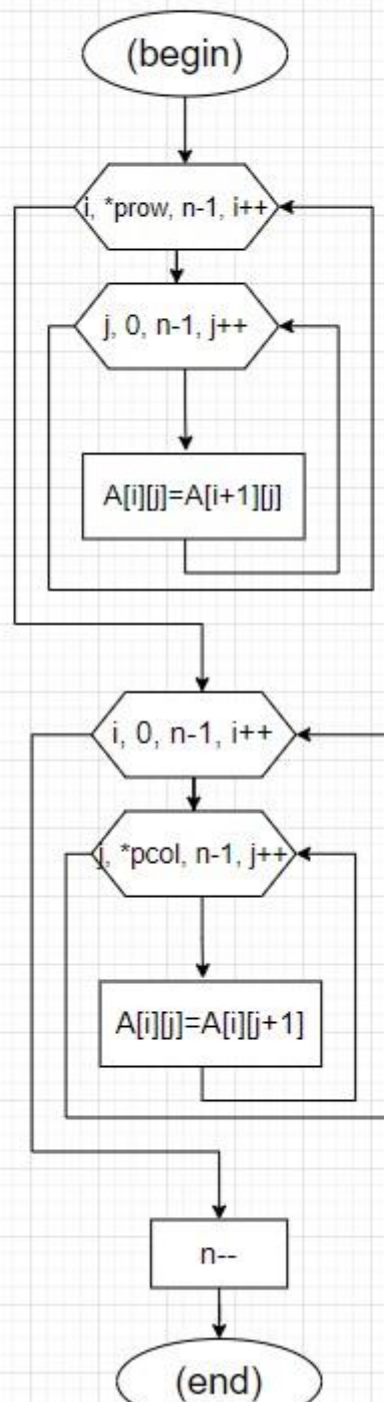
Підпрограма(3)

float abs_maximum(matrix A, int *prow, int *pcol)



алгоритма(4)

```
void delete_row_and_column(matrix A, int *prow, int *pcol)
```



Висновок: отже, програма роботи з багатовимірними масивами(матрицями) була реалізована за використанням роботи підпрограм(генерація матриці/виводи матриці/знаходження максимального за модулем елемента/видалення рядка і стовпчика). Результати отримані вірні та відповідно до умови поставленого завдання.