

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 No 8 з дисципліни

«Основи програмування»

«Багатовимірні масиви»

Варіант 13

Виконав студент Князев Ілля Сергійович
(шифр, прізвище, ім'я, по батькові)

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

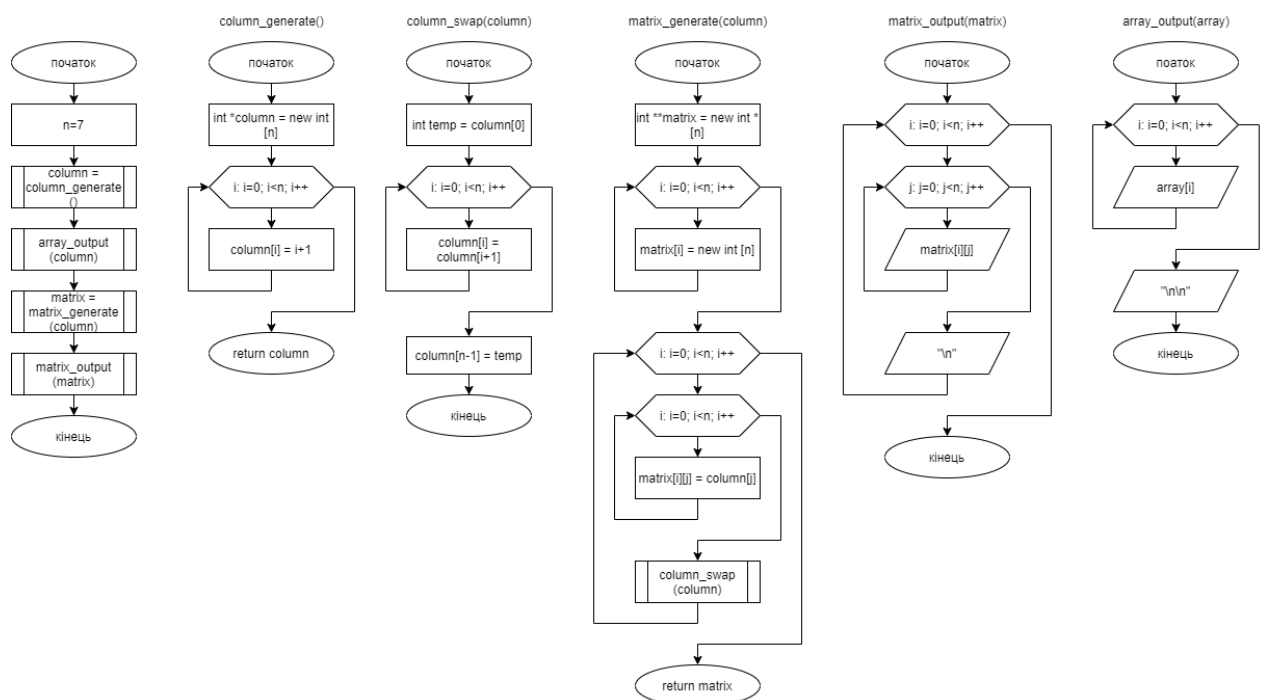
Київ 2020

Умова:

13. Дані дійсні числа a_1, \dots, a_n . Одержати квадратну матрицю порядку n , яка має наступний вигляд:

$$\begin{bmatrix} a_1 & a_2 & a_3 & \dots & a_{n-2} & a_{n-1} & a_n \\ a_2 & a_3 & a_4 & \dots & a_{n-1} & a_n & a_1 \\ a_3 & a_4 & a_5 & \dots & a_n & a_1 & a_2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_n & a_1 & a_2 & \dots & a_{n-3} & a_{n-2} & a_{n-1} \end{bmatrix}.$$

Блок-схема



Розв'язок на C++

```
#include<iostream>
#include<iomanip>
using namespace std;

const int n = 7;
```

```

int* column_generate();
int** matrix_generate(int*);
void matrix_output(int**);
void column_swap(int*);
void array_output(int*);

int main()
{
    int **matrix,
        *column;

    column = column_generate();
    array_output(column);
    matrix = matrix_generate(column);
    matrix_output(matrix);
    return 0;
}

int* column_generate()
{
    int *column = new int [n];
    for (int i=0; i<n; i++)
    {
        column[i] = i+1;
    }
    return column;
}

void column_swap(int* column)
{
    int temp = column[0];
    for (int i=0; i<n-1; i++)
        column[i] = column[i+1];
    column[n-1] = temp;
}

int** matrix_generate(int* column)
{
    int **matrix = new int *[n];
    for (int i=0; i<n; i++)
        matrix[i] = new int [n];

    for (int i=0; i<n; i++)
    {
        for (int j=0; j<n; j++)
            matrix[i][j] = column[j];
        column_swap(column);
    }
    return matrix;
}

```

```

void matrix_output(int** matrix)
{
    for (int i=0; i<n; i++)
    {
        for (int j=0; j<n; j++)
            cout << setw(4) << matrix[i][j];
        cout << '\n';
    }
}

void array_output(int *array)
{
    for (int i=0; i<n; i++)
        cout<< setw(4) << array[i];
    cout << "\n\n";
}

```

Екранна форма результатів роботи

1	2	3	4	5	6	7
1	2	3	4	5	6	7
2	3	4	5	6	7	1
3	4	5	6	7	1	2
4	5	6	7	1	2	3
5	6	7	1	2	3	4
6	7	1	2	3	4	5
7	1	2	3	4	5	6

Висновок:

Задача була проаналізована, та виконана оптимальним шляхом. Алгоритм працює при всіх допустимих вхідних даних. Задача виконана та протестована на C++.