

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»

«Дослідження алгоритмів обходу масивів »

Варіант 2

Виконав студент ПІ-15, Богун Даниїл Олександрович
(шифр, прізвище, ім'я, по батькові)

Перевірила Вечерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Київ 2021

Лабораторна робота 8

Дослідження алгоритмів пошуку та сортування

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

Варіант 2

Задача

Розробити алгоритм та написати програму, яка складається з наступних дій:

1. Опису змінної індексованого типу (двовимірний масив) згідно з варіантом .
2. Ініціювання змінної, що описана в п.1 даного завдання.
3. Обчислення змінної, що описана в п.1, згідно з варіантом .

2	Задано матрицю дійсних чисел $A[n,n]$. Знайти суму елементів, розташованих вище головній діагоналі матриці.
---	--

Постановка задачі

Створюємо двовимірний масив і заповнюємо його випадковими числами, потім шукаємо елементи, розташовані вище головної діагоналі, після чого знаходимо їхню суму.

Мат. модель

Змінна	Тип	Ім'я	Призначення
Матриця	Дійсний	matrix	Проміжне дане
Кількість рядків	Цілий	row	Початкове дане
Кількість стовпців	Цілий	col	Початкове дане
Лічильник рядків	Цілий	i	Проміжне дане
Лічильник стовпців	Цілий	j	Проміжне дане
Сума елементів, розташованих над головною діагоналлю	Дійсний	Sum	Результат
Генерація двовимірного масиву	Процедура	Matrixx	Проміжне дане
Знаходження суми елементів, розташованих над головною діагоналлю	Процедура	SumAboveDiag	Результат

Треба знайти суму елементів, розташованих вище головної діагоналі, тобто матриця квадратна (тільки у квадратних матрицях є головна та побічна діагональ), тому $row = col$.

Псевдокод алгоритму:

Основна програма:

Початок

Введення row

row = col

matrix = Matrixx(row, col)

Sum = SumAboveDiag(row, col, matrix)

Кінець

Підпрограми:

Matrixx(row, col)

Повторити

Для i від 0 до row

Повторити

Для j від 0 до col

matrix[i][j] = рандомне число

від -10 до 10

Все повторити

Все повторити

Повернути matrix

Кінець matrix

SumAboveDiag(row, col, matrix)

Повторити

Для i від 0 до row

Повторити

Для j від 0 до col

Якщо (i < j)

sum = sum + matrix[i][j];

Все повторити

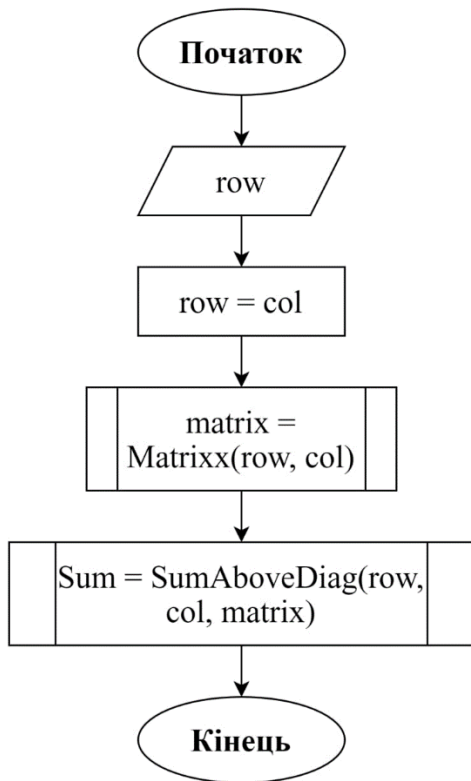
Все повторити

Повернути sum

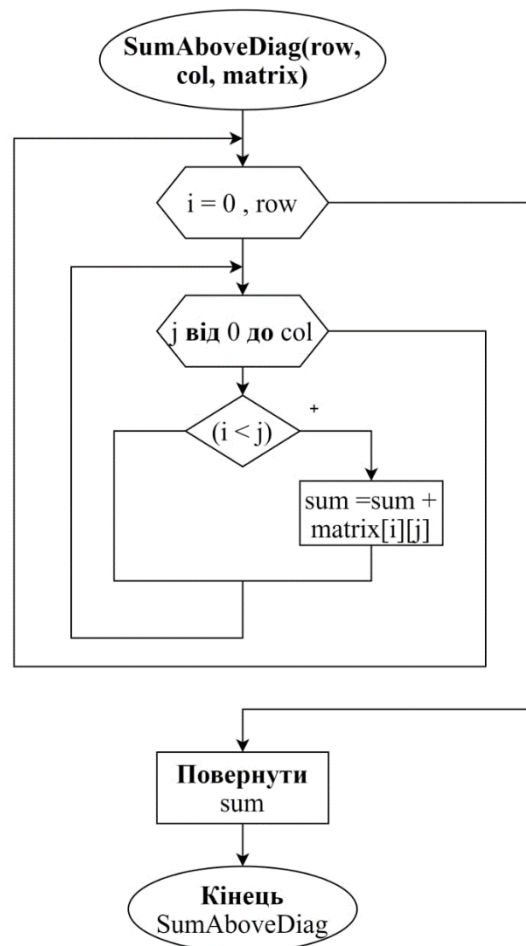
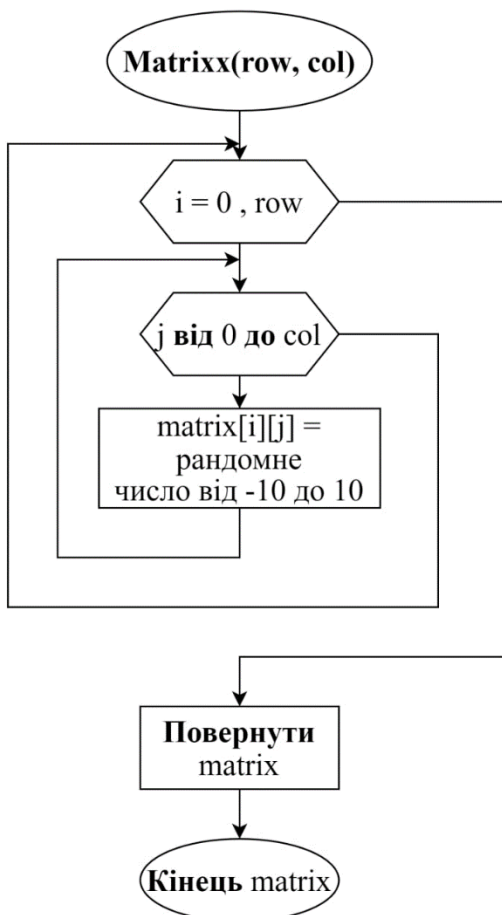
Кінець SumAboveDiag

Блок-схема алгоритму

Основна програма:



Підпрограми:



Код програми:

```
#include <iostream>
#include <iomanip>
#include <windows.h>

double SumAboveDiag(int a, int b, double** matrix);
double** Matrixx(int a, int b);
void main() {
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    srand(time(NULL));
    int row, col;
    std::cout << "The matrix (n x n) \n n = " ; std::cin >> row;
    col = row;
    double** matrix;
    matrix = Matrixx(row, col);
    double Sum = SumAboveDiag(row, col, matrix);
    std::cout << "\n\n" << " Сума елементів, розташованих над  
головною діагоналлю = " << Sum << "\n";
}

double** Matrixx(int a, int b) {
    double** arr = new double* [a];
    for (int i = 0; i < a; i++)
        arr[i] = new double[b];
    for (int i = 0; i < a; i++) {
        for (int j = 0; j < b; j++) {
            arr[i][j] = ((double)rand()) / RAND_MAX * 20 - 1 + 1 +
(-10);
            std::cout << arr[i][j] << std::setw(12);
        }
        std::cout << "\n";
    }
    return arr;
}

double SumAboveDiag(int a, int b, double** arr) {
    double sum = 0;
    for (int i = 0; i < a; i++) {
        for (int j = 0; j < b; j++) {
            if (i < j)
                sum += arr[i][j];
            else
                continue;
        }
    }
    return sum;
}
```

Тестування:

Консоль отладки Microsoft Visual Studio

The matrix (n x n)

n = 4

-4.94491	-9.45616	-7.74407	4.52681
-0.334788	-4.11115	-0.171819	7.59148
3.32438	-4.23261	-0.616169	-3.92193
2.22266	8.53145	5.09568	9.78271

Сума елементів, розташованих над головною діагоналлю = -9.1757

Консоль отладки Microsoft Visual Studio

The matrix (n x n)

n = 5

-4.7557	-6.2273	8.08161	-0.200507	-2.26112
3.57402	0.298776	-2.14209	0.722373	-0.718711
-9.19309	-1.36814	5.91052	7.94366	-8.12738
-4.89608	-4.96811	-7.65007	1.18809	5.9978
6.49342	3.54595	-3.15958	-6.60756	-4.9559

Сума елементів, розташованих над головною діагоналлю = 3.06833

Консоль отладки Microsoft Visual Studio

The matrix (n x n)

n = 8

-4.71603	4.97971	6.14978	-6.45924	3.64971	-8.49239	-1.70751	2.12439
2.28004	4.232	9.00204	-0.830409	-3.84014	9.92553	2.77322	6.75039
1.34129	4.0495	0.781579	-6.57277	4.72579	-0.169988	7.39677	-3.56243
2.40028	-2.73293	3.53618	-6.83584	5.84277	-7.47795	0.126652	-0.979949
-6.69729	-0.686361	-0.544145	0.237739	-2.93008	-9.36766	2.99417	1.08982
8.28486	-8.811	-7.79778	0.400708	-9.17051	-1.75146	-1.46153	4.04462
-3.64544	-0.936613	3.65215	6.84866	-8.94955	-6.96463	5.40269	-2.96121
-5.18235	-3.27189	-4.67635	6.35304	3.43181	-0.148625	3.15714	-1.06479
-5.6505	-9.64171	-5.91968	2.15308	6.77053	-3.2432	-5.74267	-7.55303

Сума елементів, розташованих над головною діагоналлю = 8.69533

Консоль отладки Microsoft Visual Studio

The matrix (n x n)

n = 12

-4.6025	-1.0831	7.64519	-9.29563	2.49428	8.11884	6.5746	0.284738	-3.28166	-9.66002	3.85662	0.702231
6.37257	5.57237	-5.16099	9.94385	-4.60128	-1.7185	7.22709	-8.41182	2.98868	-8.89584	2.49306	3.05826
1.34129	4.0495	0.781579	-6.57277	4.72579	-0.169988	7.39677	-3.56243	-7.05435	5.95752	9.03256	-5.67553
2.34962	-5.78234	-2.15735	-1.90466	7.50237	8.15241	-8.50581	-0.12421	6.64724	-0.826136	7.3217	2.95083
-5.91968	1.94067	-1.29917	5.12497	-7.43217	9.61364	-8.21406	1.03244	8.92087	-0.381787	-9.28404	-0.987884
5.86535	-9.45677	9.42076	-1.81738	-6.67653	0.796838	-1.8424	1.33274	-7.76482	9.63988	2.90139	9.26145
-5.95813	-8.76522	5.70544	-7.81487	8.96054	2.32337	4.16669	3.84503	6.17115	-5.83789	-1.88025	-6.07349
-0.954314	9.34202	1.67943	4.43403	-1.46214	-5.88855	8.75912	5.62548	7.86431	6.84378	-9.80529	-2.84646
1.5952	-6.75649	-9.40855	-9.98474	2.29347	8.95932	3.0253	-5.04196	-2.19214	0.177923	9.71557	-7.05252
-7.64031	2.8782	6.80593	-6.66616	-5.45396	-8.6816	1.36814	-4.65743	-4.62447	-3.57341	-1.24119	-1.49815
-0.524003	4.68368	8.44966	-2.35267	2.58278	-1.85217	6.4629	8.12006	-1.55065	6.23402	2.53151	-3.21635
8.29096	9.87609	-9.88708	4.48897	-4.10566	8.69869	7.37724	-4.39497	1.51891	-8.18171	7.29423	0.251778

Сума елементів, розташованих над головною діагоналлю = 39.8682

Висновок:

Дослідили алгоритми обходу масивів, набули практичних навичок використання цих алгоритмів під час складання програмних специфікацій.