**Лабораторная работа №4**

**студента группы ИТ-211**

**Калачева Даниила Владиславовича**

**Выполнение\_\_\_\_\_\_\_\_\_\_\_\_Защита\_\_\_\_\_\_\_\_\_\_\_\_**

**ПРИМЕНЕНИЕ ИТЕРАТИВНЫХ И РЕКУРСИВНЫХ ФУНКЦИЙ**

**Цель работы:** ознакомиться с организацией передачи параметров в функции по значению и по ссылке; получить навыки описания итеративных и рекурсивных функций.

**Содержание работы**

**Для части А:**

Разбить задачу соответствующего варианта на подзадачи таким образом, чтобы решение каждой подзадачи описывалось функцией, а основная программа состояла бы из последовательного вызова функций.

Для каждой подзадачи описать спецификацию и блок-схему алгоритма. Спецификация содержит заголовок функции, её назначение, из которого должен быть понятен смысл каждого параметра, список входных параметров, список выходных (возвращаемых) параметров.

Программа должна по выбору пользователя осуществлять ввод исходных данных с клавиатуры или из файла. В программе не должно быть глобальных переменных.

**Для части Б:**

Разработать алгоритм и составить его блок-схему и программу для решения задачи соответствующего варианта двумя способами: используя рекурсивную и итеративную функции. В программе не должно быть глобальных переменных.

**Вариант №7**

**Часть А**

Дана квадратная матрица. Назовем псевдодиагональю множество элементов этой матрицы, лежащих на прямой, параллельной прямой, содержащей элементы ai, i. Найти сумму максимальных элементов всех псевдодиагоналей данной матрицы.

**Часть Б**

Найти номер первого вхождения максимального значения в последовательность натуральных чисел длины n. Последовательность не хранить.

**Ход работы**

**Часть А**

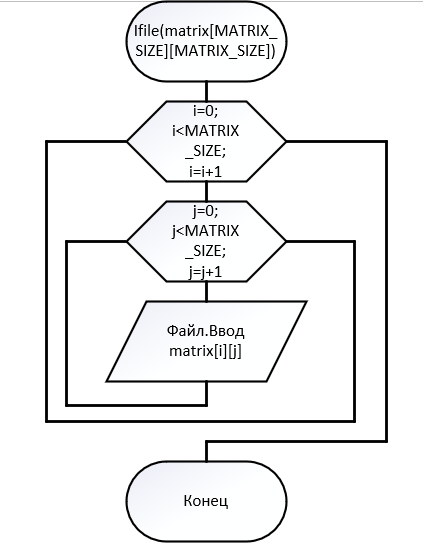
1. Для решения поставленной задачи, составил блок-схемы функций и их спецификации. (см. Рисунок 1, Рисунок 2, Рисунок 3, Рисунок 4, Рисунок 5)

**Функция ifile** (см. Рисунок 1)

**Назначение:** ввод из файла

**Входные данные**: двумерный массив фиксированного размера

**Выходные данные**: отсутствуют



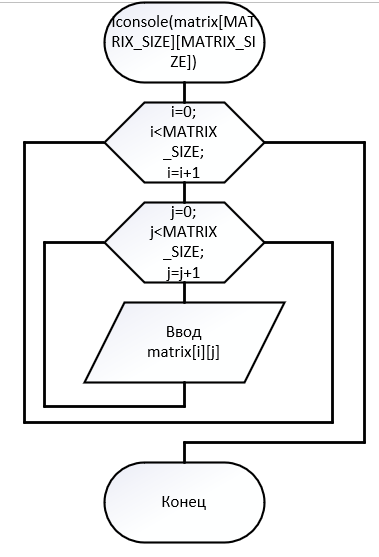
1. Функция ifile

**Функция iconsole** (см. Рисунок 2)

**Назначение:** ввод из консоли

**Входные данные**: двумерный массив фиксированного размера

**Выходные данные**: отсутствуют



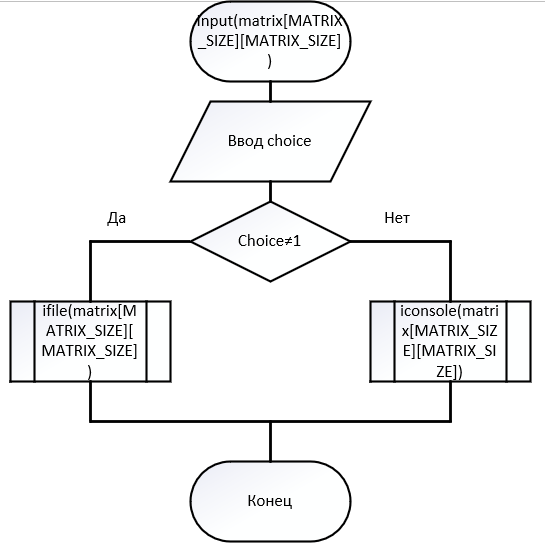
1. Функция iconsole

**Функция input** (см. Рисунок 3)

**Назначение:** выбор ввода и вызов функций ввода

**Входные данные**: двумерный массив фиксированного размера

**Выходные данные**: отсутствуют



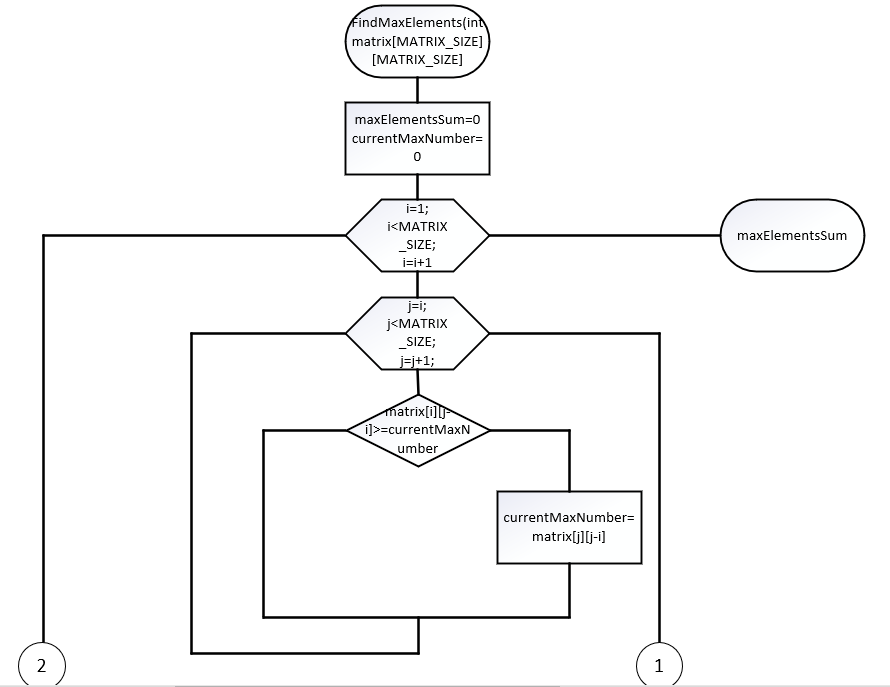
1. Функция input

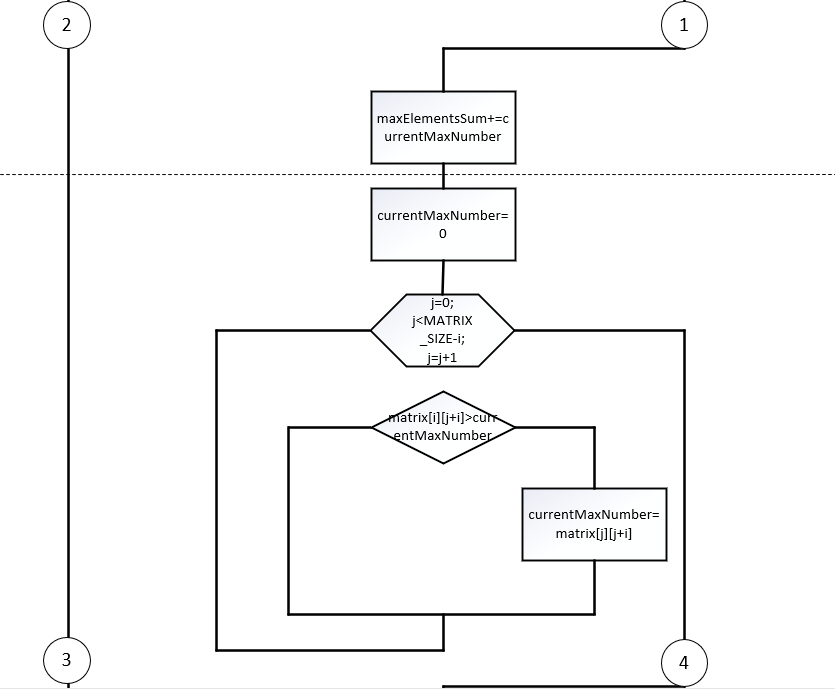
**Функция FindMaxElements** (см. Рисунок 4)

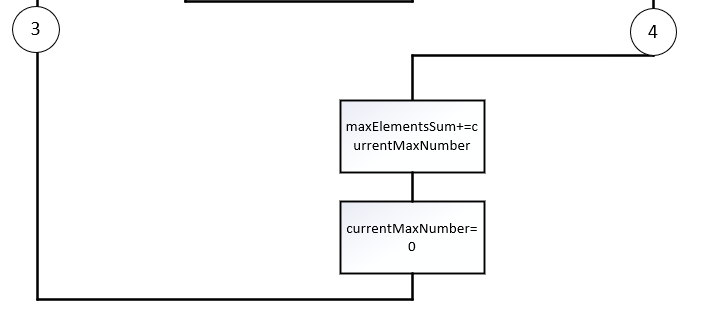
**Назначение:** вычислить сумму максимальных элементов псевдодиагоналей

**Входные данные**: двумерный массив фиксированного размера

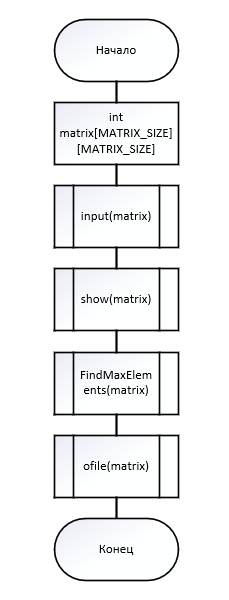
**Выходные данные**: число, которое является суммой максимальных элементов псевдодиагоналей







1. Функция findmax



1. Основная программа
2. Написал код программы на языке С++.

#include <iostream>

#include <fstream>

#include <locale>

#define MATRIX\_SIZE 3

using namespace std;

void ifile(int matrix[MATRIX\_SIZE][MATRIX\_SIZE])

{

ifstream input("Lab4.txt");

for (int i = 0; i < MATRIX\_SIZE; i++)

for (int j = 0; j < MATRIX\_SIZE; j++)

input >> matrix[i][j];

input.close();

}

void iconsole(int matrix[MATRIX\_SIZE][MATRIX\_SIZE])

{

for (int i = 0; i < MATRIX\_SIZE; i++)

for (int j = 0; j < MATRIX\_SIZE; j++)

{

cout << "Введите " << j + 1 << " элемент " << i + 1 << " строки матрицы: ";

cin >> matrix[i][j];

}

}

void input(int matrix[MATRIX\_SIZE][MATRIX\_SIZE])

{

int choice;

cout << "Введите 1 для ввода из консоли, иначе (файл. ввод) - любое другое число" << endl;

cout << "Создайте или измените файл Lab4.txt и в него запишите по порядку элементы матрицы размером " << MATRIX\_SIZE << "х" << MATRIX\_SIZE << " через пробел" << endl;

cin >> choice;

if (choice != 1)

ifile(matrix);

else

iconsole(matrix);

}

int FindMaxElements(int matrix[MATRIX\_SIZE][MATRIX\_SIZE])

{

int maxElementsSum = 0;

int currentMaxNumber = 0;

for (int i = 1; i < MATRIX\_SIZE; i++)

{

for (int j = i; j < MATRIX\_SIZE; j++)

{

if (matrix[j][j - i] >= currentMaxNumber)

currentMaxNumber = matrix[j][j - i];

}

maxElementsSum += currentMaxNumber;

currentMaxNumber = 0;

for (int j = 0; j < MATRIX\_SIZE - i; j++)

{

if (matrix[j][j + i] > currentMaxNumber)

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

}

maxElementsSum += currentMaxNumber;

currentMaxNumber = 0;

}

return maxElementsSum;

}

void Show(int matrix[MATRIX\_SIZE][MATRIX\_SIZE])

{

cout << endl << endl;

for (int i = 0; i < MATRIX\_SIZE; i++)

{

for (int j = 0; j < MATRIX\_SIZE; j++)

{

cout << matrix[i][j] << "\t";

}

cout << endl;

}

}

void ofile(int matrix[MATRIX\_SIZE][MATRIX\_SIZE])

{

ofstream file("Lab4aOutput.txt");

file << FindMaxElements(matrix);

}

int main()

{

setlocale(LC\_ALL, "");

int matrix[MATRIX\_SIZE][MATRIX\_SIZE];

input(matrix);

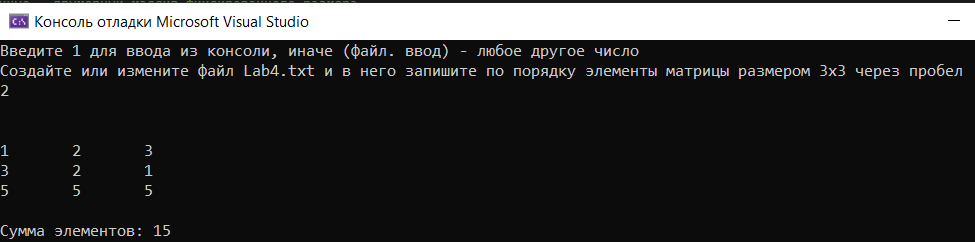
Show(matrix);

cout << endl << "Сумма элементов: " << FindMaxElements(matrix);

ofile(matrix);

}

1. Результаты работы программы. (см. Рисунок 6)



1. Результаты работы программы

**Часть Б**

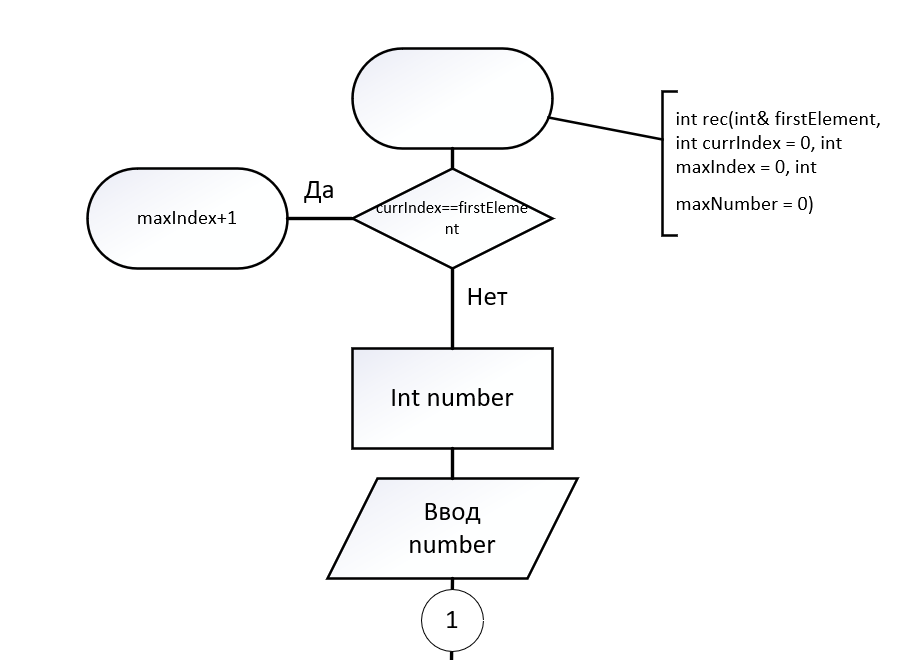
1. Составил блок-схемы и спецификации для решения задачи, соответствующей моему варианту.

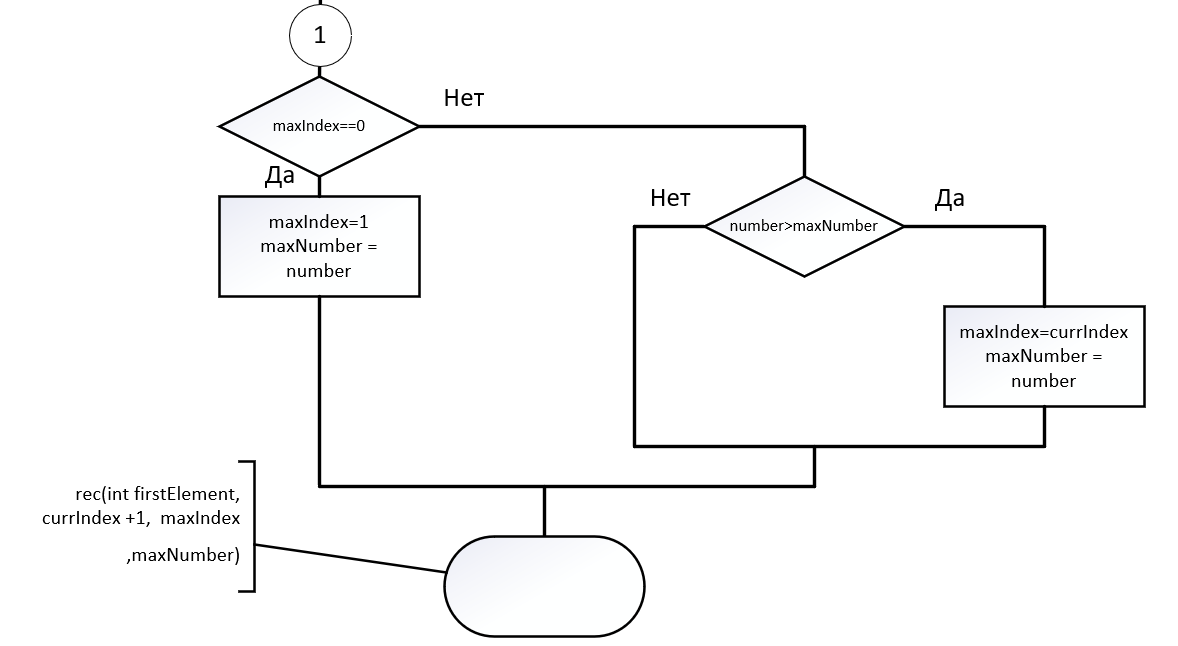
**Рекурсивная функция rec** (см. Рисунок 7)

**Назначение:** определить индекс первого вхождения максимального элемента последовательности с помощью рекурсии.

**Входные данные**: целое число (элемент последовательности)

**Выходные данные**: индекс первого вхождения максимального элемента последовательности





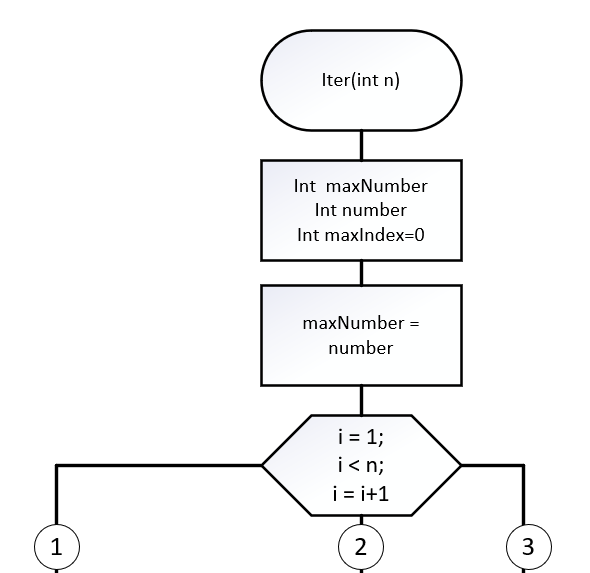
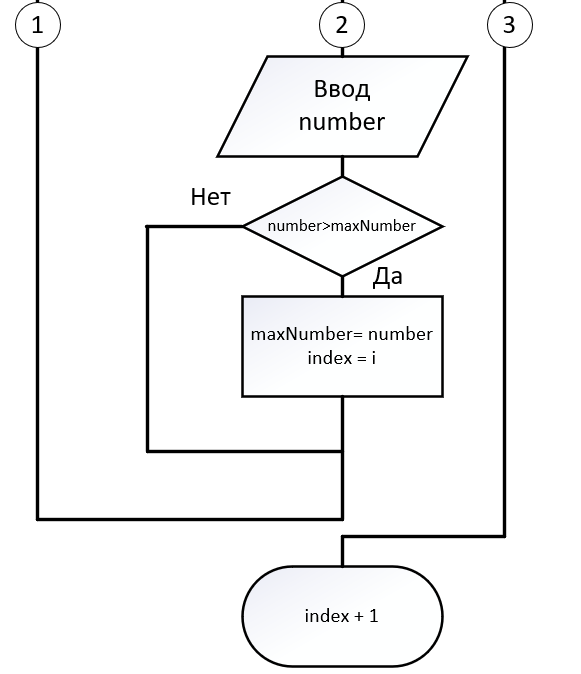
1. Рекурсивная функция rec

**Итеративная функция iter** (см. Рисунок 8)

**Назначение:** определить индекс первого вхождения максимального элемента последовательности с помощью итерации

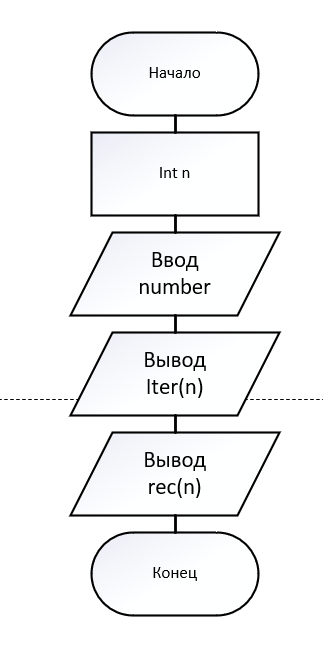
**Входные данные**: целое число (элемент последовательности)

**Выходные данные**: индекс первого вхождения максимального элемента последовательности (счет с нуля)

****

1. Итеративная функция iter

Основная программа для итеративной и рекурсивной функции. (см. Рисунок 9)



1. Основная программа для итерации и рекурсии
2. Написал код программы на языке С++. Итеративный и рекурсивный алгоритм:

#include <iostream>

#include <locale>

using namespace std;

int rec(int& firstElement, int currIndex = 0, int maxIndex = 0, int maxNumber = 0)

{

if (currIndex == firstElement)

return maxIndex + 1;

int number;

cout << "Введите число # " << currIndex + 1 << ": ";

cin >> number;

if (maxIndex == 0)

{

maxIndex = 1;

maxNumber = number;

}

else if (number > maxNumber)

{

maxIndex = currIndex;

maxNumber = number;

}

return rec(firstElement, currIndex + 1, maxIndex, maxNumber);

}

int iter(int n)

{

int maxNumber;

int number;

int index = 0;

cout << "Введите число # " << 1 << ": ";

cin >> number;

maxNumber = number;

for (int i = 1; i < n; i++)

{

cout << "Введите число # " << i + 1 << ": ";

cin >> number;

if (number > maxNumber)

{

maxNumber = number;

index = i;

}

}

return index + 1;

}

int main()

{

setlocale(LC\_ALL, "ru");

int n;

cout << "Введите длину последовательности: ";

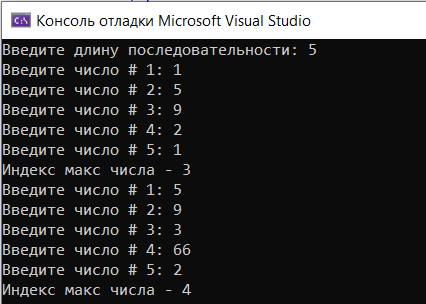
cin >> n;

cout << "Индекс макс числа - " << iter(n) << endl;

cout << "Индекс макс числа - " << rec(n) << endl;

}

1. Проверим работоспособность обоих алгоритмов. (см. Рисунок 10)



1. Работоспособность итеративной и рекурсивной функции

**Вывод:** изучил способы создания функций, ознакомился с организацией передачи параметров в функции по значению, получил навыки описания итеративных и рекурсивных функций.