***Отчет по лабораторной работе №3***

***Выполнил студент группы 1-41 xх***

***Прохоров Максим***

**Содержание**

Условие задачи \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 3

Математическая модель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3

Алгоритм решения задачи \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3

Программа\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_4

Тестовые примеры\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_4

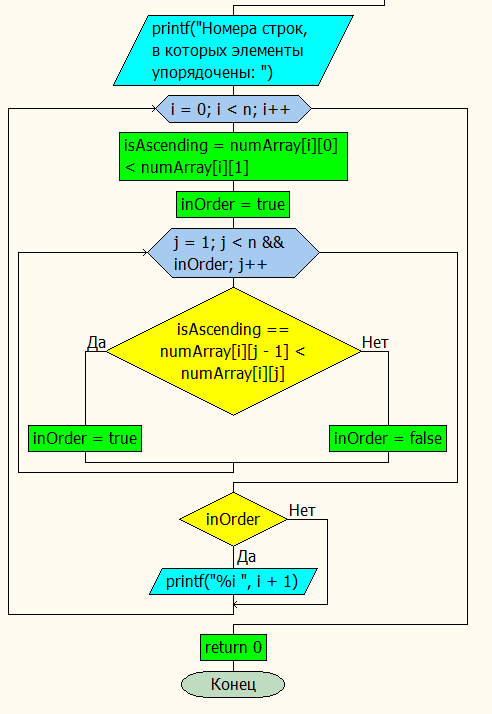
**Условие задачи**

*Вариант 26. Упорядоченные строки.* Дана целочисленная квадратная матрица порядка n. Найти номера строк, элементы каждой из которых образуют упорядоченную последовательность.

**Математическая модель**

Вводим двумерный массив. Проходимся по каждой строке: берем 2 первых элемента и определяем, возрастать или убывать должна последовательность. Далее проходимся по всем элементам строки и рассматриваем все соседние элементы: если при сравнении элементов получаем соответствие порядку всей строки, то ставим флаг inOrder = true и переходим к следующей итерации; иначе ставим inOrder = false и выходим из всего цикла. Если после цикла флаг inOrder == true, то выводим индекс текущей строки.

**Алгоритм решения задачи**



**Программа**

#include <iostream>

#include <stdio.h>

#include <locale.h>

int main()

{

setlocale(LC\_ALL, "rus");

const int n = 3;

printf("Массив порядка %i\n", n);

int numArray[n][n];

for (int i = 0; i < n; i++) {

printf("Введите элементы %i сторки: ", i+1);

for (int j = 0; j < n; j++) {

scanf\_s("%d", &numArray[i][j]);

}

}

printf("Номера строк, в которых элементы упорядочены: ");

for (int i = 0; i < n; i++) {

bool isAscending = numArray[i][0] < numArray[i][1];

bool inOrder = true;

for (int j = 1; j < n && inOrder; j++) {

if (isAscending == numArray[i][j - 1] < numArray[i][j]) {

inOrder = true;

}

else {

inOrder = false;

}

}

if (inOrder) {

printf("%i ", i + 1);

}

}

return 0;

}

**Тестовые примеры**

|  |  |
| --- | --- |
| ***Введенные данные*** | ***Результат программы*** |
| 1 2 3  3 2 1  3 3 3 | 1 2 |
| 12 23 12  9 9 9  2 3 4 | 3 |
| 85 3 2  9 12 45  8 4 9 | 1 2 |