**ЛАБОРАТОРНАЯ РАБОТА №2**

**АЛГОРИТМЫ ОБРАБОТКИ МАССИВОВ**

***Задача 1.5.2.9***

***Условие задачи***

В приведенных ниже задачах константы *A, B* (если они есть в варианте) и вектора *X* и (или) *Y* задаются вводом; *n* — размер каждого из векторов *X* и *Y*.

Найти число и произведение положительных элементов вектора *X*, удовлетворяющих требованию sin *Xk ≤* 0.

***Состав данных***

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя** | **Смысл** | **Тип** | **Структура** |
| Исходные данные | | | |
| **n** | число элементов массива | целый | простая переменная |
| **а** | заданный массив | целый | одномерный массив из 100 элементов |
| Выходные данные | | | |
| **Mult** | произведение элементов массива, удовлетворяющих условию | веществ. | простая переменная |
| **Sum** | Сумма элементов массива, удовлетворяющих условию | веществ. | Простая переменная |
| Промежуточные данные | | | |
| **i** | счетчик циклов | целый | простая переменная |
| **Flag** | флаг автоматического построения массива | символьный | простая переменная |

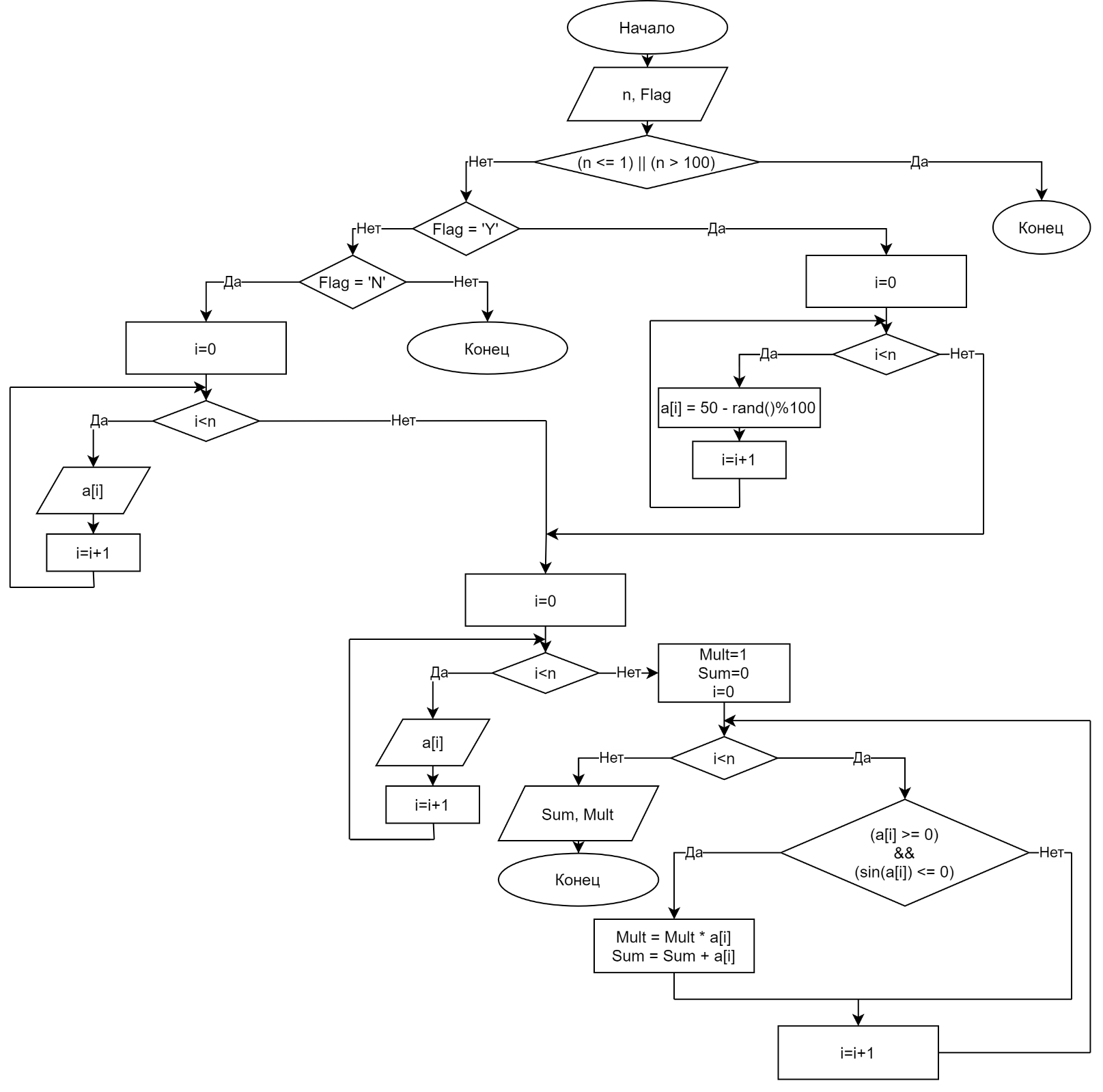
***Форма ввода***

|  |
| --- |
| Введите размер массива:  <n>  Составить массив автоматически? (Y(Да)/N(Нет))  <Flag>  Введите поочередно элементы массива:  <a[0]> <a[1]> … <a[n]> |

***Форма вывода***

|  |
| --- |
| Массив:  <a[0]> <a[1]> … <a[n]>  Произведение элементов, удовлетворяющих условию: <Mult>  Сумма элементов, удовлетворяющих условию: <Sum> |

***Блок-схема алгоритма***

******

***Программа на СИ (консольное приложение MS Visual Studio 2008)***

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

setlocale(0, "");

srand(static\_cast<unsigned int>(time(0))); // Используем системные часы для начальной генерации рандомных чисел

int a[100];

cout << "Введите размер массива: " << endl;

int n;

cin >> n;

try

{

if ((n <= 1) || (n > 100))

throw "Количество элементов массива должно быть в диапазоне от 2 до 100";

cout << "Составить массив автоматически? (Y(Да)/N(Нет))" << endl;

char Flag;

cin >> Flag;

try

{

if ((Flag != 'N') && (Flag != 'Y'))

throw "Необходимо выбрать Y-Да N-Нет";

if (Flag == 'Y') {

for (int i = 0; i < n; i += 1) {

a[i] = 50 - rand()%100;

}

}

if (Flag == 'N') {

cout << "Введите поочередно элементы массива: " << endl;

for (int i = 0; i < n; i += 1) {

cin >> a[i];

}

}

cout << "Массив:" << endl;

for (int i = 0; i < n; i += 1) cout << a[i] << "\t";

cout << endl;

int Mult=1;

int Sum=0;

for (int i = 0; i < n; i += 1) {

if ((a[i] >= 0) && (sin(a[i]) <= 0)) {

Mult = Mult \* a[i];

Sum = Sum + a[i];

}

}

cout << "Произведение элементов, удовлетворяющих условию: " << Mult << endl << "Сумма элементов, удовлетворяющих условию: " << Sum << endl;

}

catch (const char\* exception)

{

cerr << "Ошибка: " << exception << endl;

}

}

catch (const char\* exception)

{

cerr << "Ошибка: " << exception << endl;

}

system("pause");

return 0;

}

***Задача 2.2.3.9***

***Условие задачи***

Дана квадратная матрица *A* размером *n* и (или) вектор *X* из *n* элементов. Способ задания *n* и *m* определяется средой программиро­вания и указаниями преподавателя. Рекомендуется, чтобы алгоритм решения содержал не более одного двукратного цикла обработки.

Получить вектор *C* по правилу: *Ck* = 1, если каждый элемент k-го столбца, кроме первого и последнего элементов, меньше суммы пре­дыдущего и последующего элементов столбца, иначе *Ck* = 0.

***Состав данных***

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя** | **Смысл** | **Тип** | **Структура** |
| Исходные данные | | | |
| **A** | исходный массив | целый | Двумерный массив из 50×50 элементов |
| **n** | число элементов массива | целый | простая переменная |
| Выходные данные | | | |
| **C** | вектор, составленный согласно условию задачи | целый | одномерный массив из 50 элементов |
| Промежуточные данные | | | |
| **i** | счетчик циклов | целый | простая переменная |
| **j** | счетчик циклов | целый | простая переменная |
| **create\_flag** | флаг автоматического построения массива | символьный | простая переменная |
| **flag** | счетчик выполнения условия в столбце | целый | простая переменная |

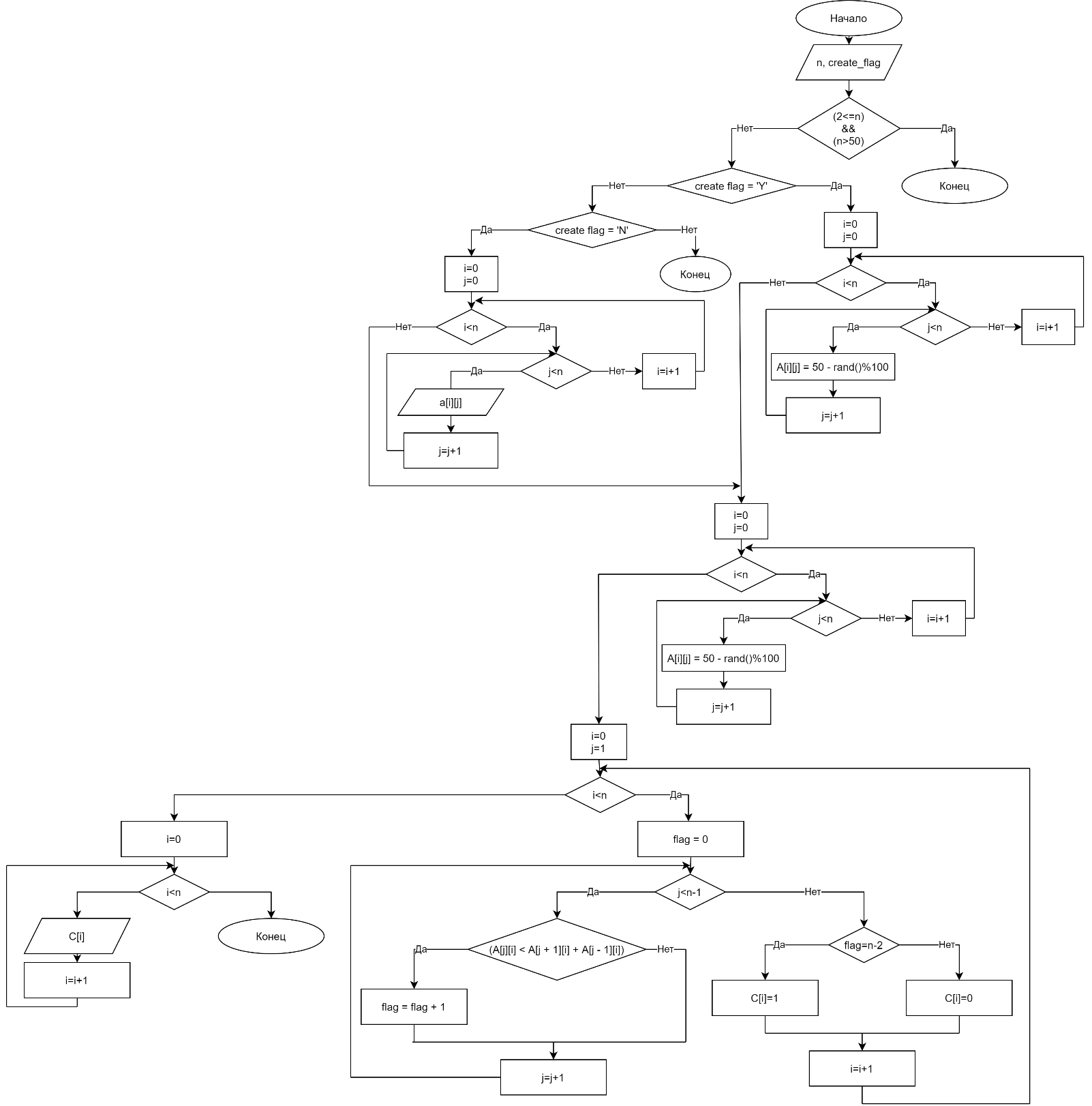
***Форма ввода***

|  |
| --- |
| Введите n - размер квадратной матрицы А  <n>  Создать матрицу автоматически? (Y-да/N-нет)  <create\_flag>  Введите элементы массива:  Cтрока № <i+1>:  <a[1][1]>  <a[1][2]>  …  <a[1][n]>  …  Строка № <i+1>:  <a[n][1]>  <a[n][2]>  …  <a[n][n]> |

***Форма вывода***

|  |
| --- |
| Полученный массив:  <A[1][1]> <A[1][2]> … <A[1][n]>  <A[2][1]> <A[2][2]> … <A[2][n]>  …  <A[n][1]> <A[n][2]> … <A[n][n]>  Полученный вектор:  <C[1]> <C[2]> … <C[n]> |

***Блок-схема алгоритма:***

******

***Программа на СИ (консольное приложение MS Visual Studio 2008)***

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "Russian");

srand(static\_cast<unsigned int>(time(0))); // Привязка к времени системных часов для получения "рандомных" чисел

int A[50][50], C[50];

int n;

cout << "Введите n - размер квадратной матрицы А" << endl;

cin >> n;

try

{

if ((n <= 2) || (n > 50))

throw "Значение n должно быть в диапазоне от 3 до 50!";

cout << "Создать матрицу автоматически? (Y-да/N-нет)" << endl;

char create\_flag;

cin >> create\_flag;

try

{

if ((create\_flag != 'Y') && (create\_flag != 'N'))

throw "Необходимо выбрать Y-да/N-нет!";

// Автоматическое создание матрицы

if (create\_flag == 'Y') {

for (int i = 0; i < n; i += 1) {

for (int j = 0; j < n; j += 1) {

A[i][j] = 50 - rand()%100;

}

}

}

// Ручное создание матрицы

if (create\_flag == 'N') {

cout << "Введите элементы массива:" << endl;

for (int i = 0; i < n; i += 1) {

cout << "Строка №" << i+1 << ":" << endl;

for (int j = 0; j < n; j += 1) {

cin >> A[i][j];

}

}

}

// Вывод массива

cout << "Полученный массив: " << endl;

for (int i = 0; i < n; i += 1) {

for (int j = 0; j < n; j += 1) {

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

}

cout << endl;

}

// Выполнение условия создания ветора C

for (int i = 0; i < n; i += 1) {

int flag = 0;

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

if (A[j][i] < A[j + 1][i] + A[j - 1][i]) flag = flag + 1;

}

if (flag == n - 2) C[i] = 1;

else C[i] = 0;

}

// Вывод вектора C

cout << "Полученный вектор:" << endl;

for (int i = 0; i < n; i += 1) cout << C[i] << "\t";

cout << endl;

}

catch (const char\* exception)

{

cerr << "Ошибка: " << exception << '\n';

}

}

catch (const char\* exception)

{

cerr << "Ошибка: " << exception << '\n';

}

system("pause");

return 0;

}