

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»

КАФЕДРА ТЕОРЕТИЧЕСКОЙ И ПРИКЛАДНОЙ ИНФОРМАТИКИ

Расчетно-графическое задание
по дисциплине «Основы программирования»

Факультет: ПМИ

Группа: ПМИ-03

Студент: Сидоров Д. И.

Преподаватель: Еланцева Е. Л.

Условие задачи: Задано множество точек на плоскости, не лежащих на одной прямой. Определить минимальное подмножество точек, после удаления которых остаются точки, лежащие на одной прямой.

1) *Анализ данных:*

Входные данные: Количество точек(не менее 2), координаты точек.

Выходные данные: Минимальное подмножество точек, после удаления которых остаются точки, лежащие на одной прямой.

Метод решения: Пользователь вводит число точек. Запускается подзадача **error**, которая проверит правильность входных данных. Объявим динамический двумерный массив, в котором будут записаны координаты точек. Далее запускается подзадача **input**, при помощи которой осуществляется ввод координат. Выведем координаты x и у, при помощи подзадач **outputX** и **outputY**. Далее запускается подзадача **formula**, в которой при помощи формулы $(x_3-x_1)*(y_2-y_1)-(x_2-x_1)*(y_3-y_1)$ (формула определяет принадлежность трех точек одной прямой) определяется прямая, содержащая максимальное количество точек. Если условие не сработало хотя бы один раз, то, по умолчанию, прямая будет состоять из двух точек, так как для построения прямой необходимы любые две точки. Далее из общего количества точек мы вычтем количество точек, лежащих на самой длинной прямой. Полученное число точек и будет минимальном подмножеством точек, после удаления которых останутся точки, лежащие на одной прямой. Если все точки лежат на одной прямой, программа выведет 0.

Можно выделить подзадачи:

error – проверяет, чтобы количество точек было не меньше двух.

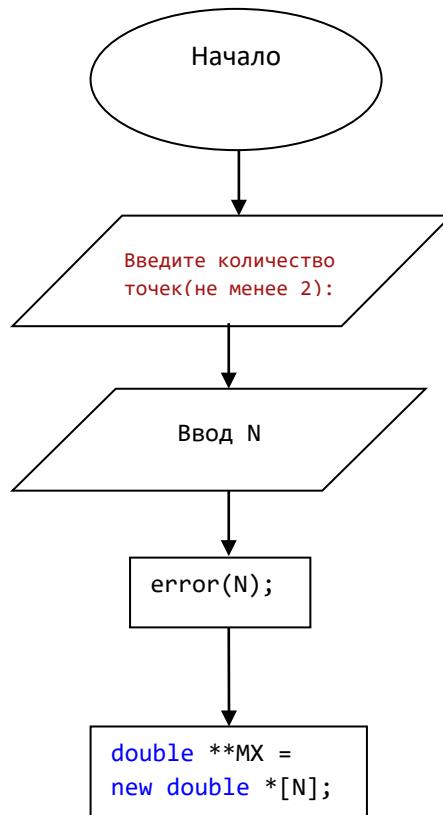
input – подпрограмма для ввода в динамический двумерный массив координат точек.

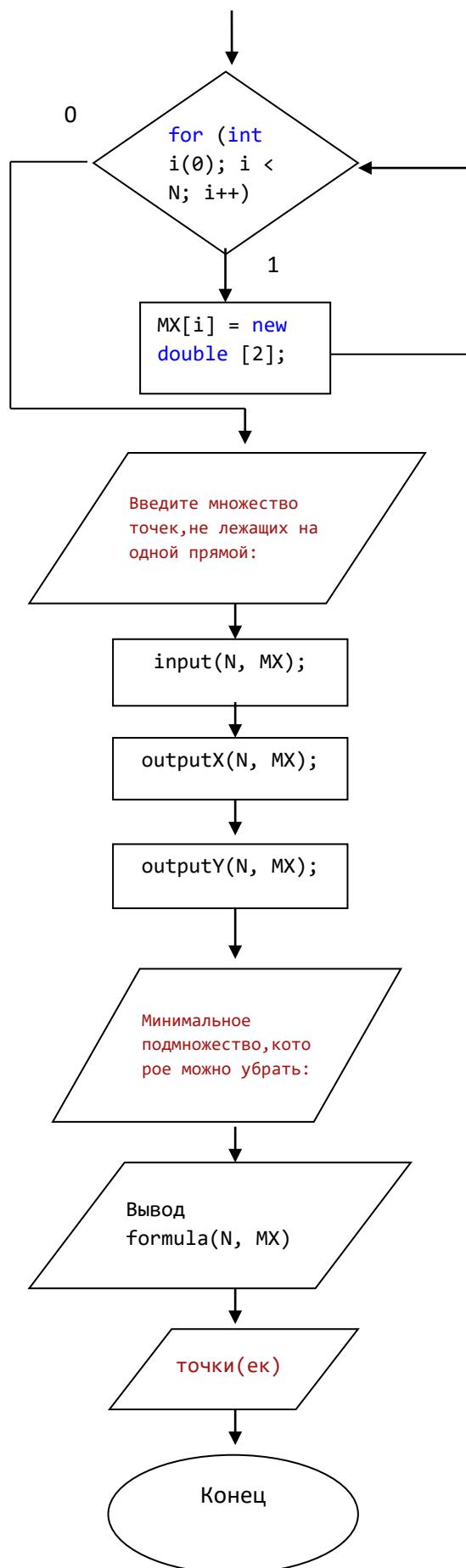
outputX – вывод координаты x.

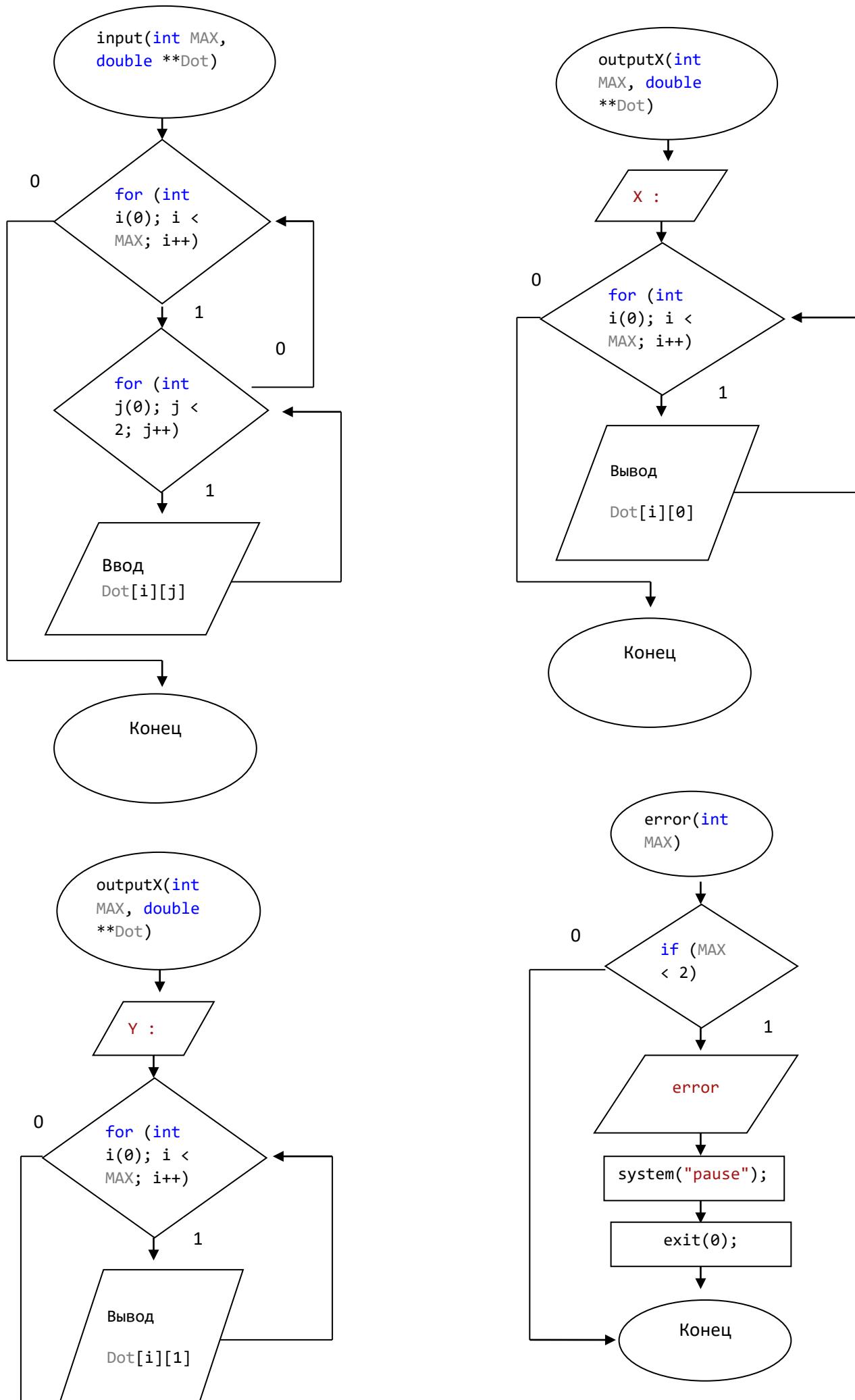
outputY – вывод координаты y.

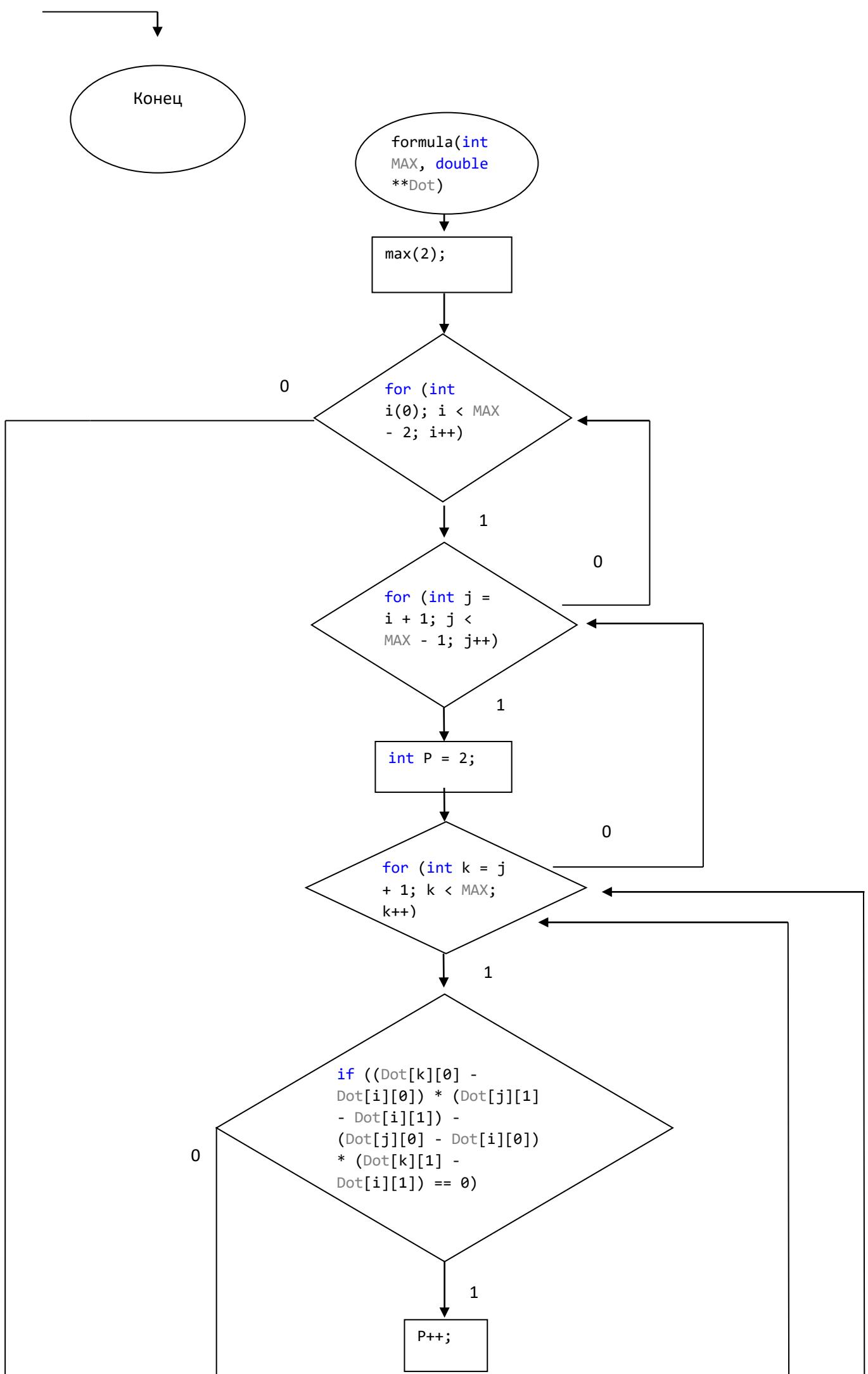
formula – определяет прямую, содержащую максимальное количество точек.

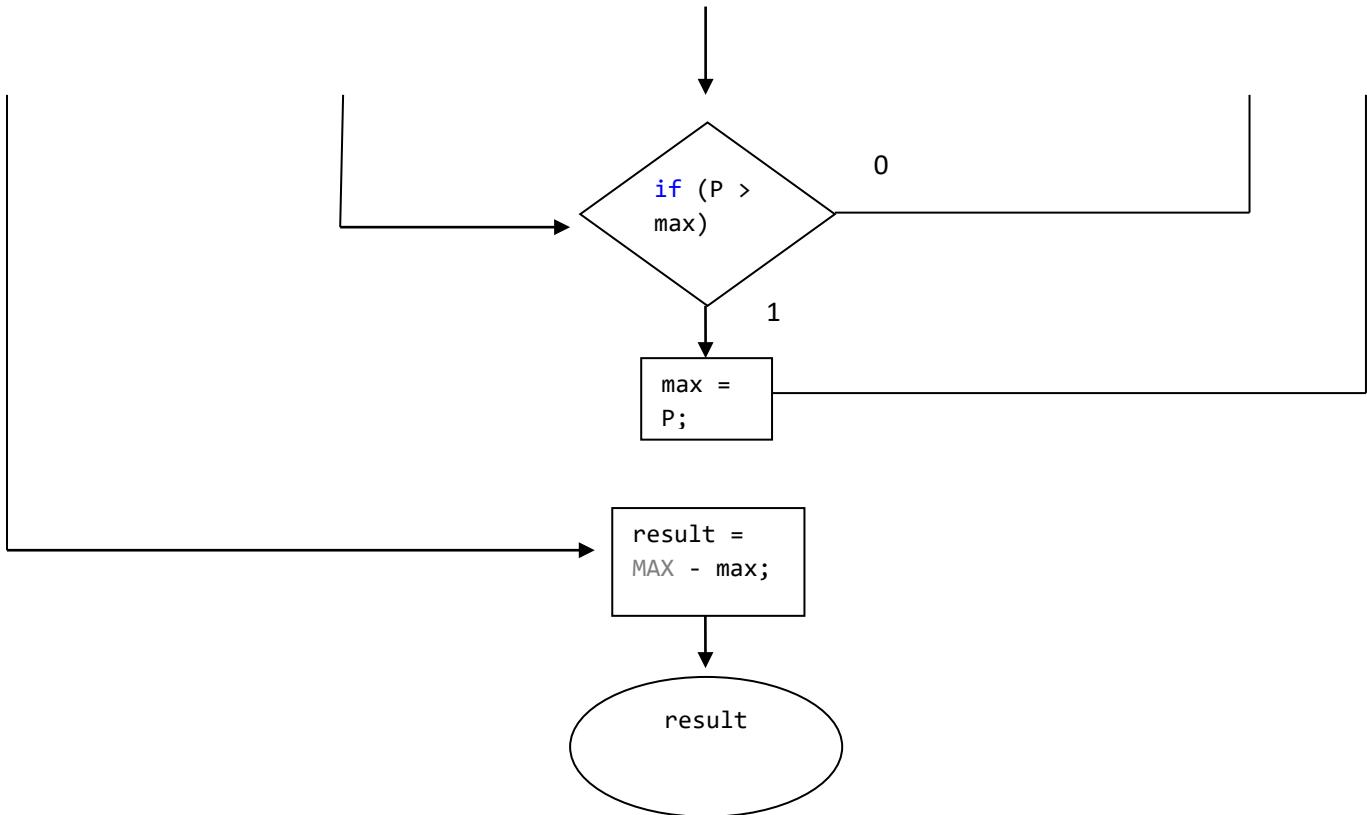
2) *Алгоритм:*



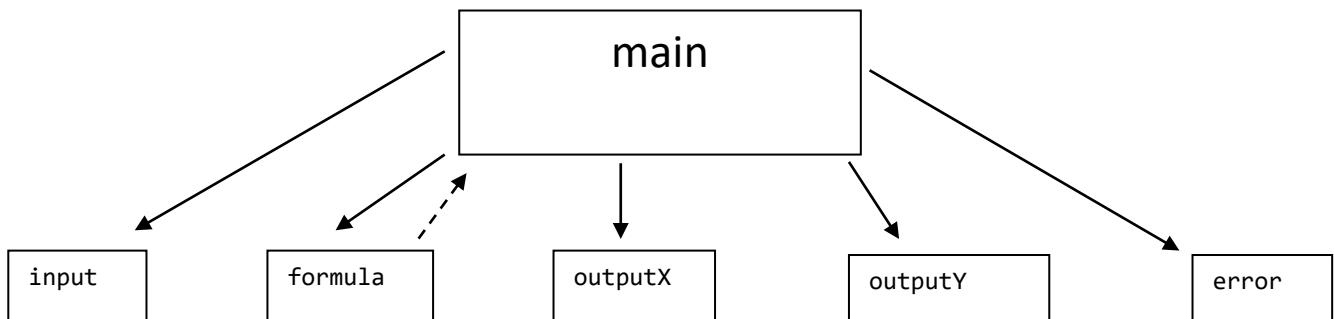








4)Структура программы



1)Процедура ввода элементов массива:

`input(int MAX, double **Dot)`

Входные данные: MAX – количество точек.

`**Dot` – массив с координатами.

2)Процедура вывода элементов массива:

`outputX(int MAX, double **Dot)`

Входные данные: MAX – количество точек.

`**Dot` – массив с координатами.

3)Процедура вывода элементов массива:

`outputY(int MAX, double **Dot)`

Входные данные: MAX – количество точек.

`**Dot` – массив с координатами.

4)Процедура проверки входных данных

`error(int MAX)`

Входные данные: MAX – количество точек.

5)Функция вычисления прямой, содержащей максимальное число точек.

formula(**int** MAX, **double** **Dot)

Входные данные: MAX – количество точек.

**Dot – массив с координатами.

Выходные данные: result – количество лишних точек.

5)Текст программы:

```
#include<iostream>
using namespace std;
void input(int MAX, double **Dot) {
    for (int i(0); i < MAX; i++) {
        for (int j(0); j < 2; j++) {
            cin >> Dot[i][j];
        }
    }
}
int formula(int MAX, double **Dot) {
    int result;
    int max(2);
    for (int i(0); i < MAX - 2; i++) {
        for (int j = i + 1; j < MAX - 1; j++) {
            int P = 2;
            for (int k = j + 1; k < MAX; k++) {
                if ((Dot[k][0] - Dot[i][0]) * (Dot[j][1] - Dot[i][1]) - (Dot[j][0] - Dot[i][0]) * (Dot[k][1] - Dot[i][1]) == 0) {
                    P++;
                }
                if (P > max) max = P;
            }
        }
    }
    result = MAX - max;
    return result;
}
void outputX(int MAX, double **Dot){
    cout << "X : ";
    for (int i(0); i < MAX; i++) {
        cout << Dot[i][0] << " ";
    }
    cout << endl;
}
void outputY(int MAX, double **Dot) {
    cout << "Y : ";
    for (int i(0); i < MAX; i++) {
        cout << Dot[i][1] << " ";
    }
    cout << endl;
}
void error(int MAX) {
    if (MAX < 2)
    {
        cout << "error" << endl;
        system("pause");
        exit(0);
    }
    cout << endl;
}
int main()
{
    setlocale(LC_ALL, "rus");
    int N;
    cout << "Введите количество точек(не менее 2): " << endl;
    cin >> N;
    error(N);
    double **MX = new double *[N];
    for (int i(0); i < N; i++)
    {
        MX[i] = new double[2];
```

```

    }
    cout << "Введите множество точек, не лежащих на одной прямой:" << endl;
    input(N, MX);
    outputX(N, MX);
    outputY(N, MX);
    cout << "Минимальное подмножество, которое можно убрать: " << formula(N, MX) << " точки(ек)";
}

```

6) Тесты:

№	Дано		Результат	Примечание
	Количество точек	Координаты точек		
1	2	X 1 2 Y 2 3	0	Прямая состоит из минимального возможного количества точек
2	1	X 1 Y 1	error	Нарушено условие ввода
3	3	X 1 2 3 Y 1 2 4	1	Работает правильно
4	6	X 1 2 3 4 5 6 Y 1 2 3 5 7 8	3	Работает правильно
5	5	X 1 2 3 4 4 Y 1 2 3 5 6	2	Две прямые, одна содержит больше точек, чем другая.
6	5	X 1 2 3 4 5 Y 1 2 3 4 5	0	Все точки лежат на одной прямой
7	3	X 1.1 2.2 3.5 Y 1.1 2.3 3.5	1	Вещественные числа

7) Результат работы программы:

Программа работает правильно, что подтверждают тесты.