МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»

РУТ(МИИТ)

Кафедра “УТБиИС”

**Отчёт №8**

**по дисциплине Информатика**

**Вариант 16**

Выполнил: Котов А.А.

Группа: УМЛ-112

Проверил: Новиков А.И.

МОСКВА 2020

**Задание**

Овцы пасутся примерно вместе. Отбившейся от стада считается овца, которая удалилась на максимальное расстояние от условного “центра стада”. Если известны координаты всех овец (как точек на плоскости), проверить, если ты овца, отбившаяся от стада. Она достанется волку, который с ноутбуком сидит под ближайшим к стаду кустом. Использваоть функции обработки массивов.

**Код на языке C++**

#include <iostream>

#include <iomanip>

using namespace std;

//вывести веса

void write\_mas(double\* m, int size)

{

for (int i = 0; i < size; i++)

{

cout << setw(4) << setprecision(2) << m[i] << " ";

}

cout << "\n";

}

double\* randomize(int n)

{

double\* vals = new double[n]{0};

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

{

vals[i] = (rand() % 100 - 50) / 10.; // [-5..5]

}

return vals;

}

double average(double\* m, int size)

{

double av = 0;

for (int i = 0; i < size; i++)

{

av += m[i];

}

av /= size;

return av;

}

double max\_dst\_index(double\* x, double\* y, int size, double c\_x, double c\_y)

{

double max\_dst = sqrt(pow(x[0] - c\_x, 2) + pow(y[0] - c\_y, 2));

int max\_index = 0;

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

{

double dst = sqrt(pow(x[i] - c\_x, 2) + pow(y[i] - c\_y, 2));

if (dst > max\_dst)

{

max\_dst = dst;

max\_index = i;

}

}

return max\_index;

}

double\* remove\_at(double\* m, int size, int index)

{

double\* r = new double[size - 1];

for (int i = 0; i < index; i++)

{

r[i] = m[i];

}

for (int i = index + 1; i < size; i++)

{

r[i - 1] = m[i];

}

delete[] m;

return r;

}

int main()

{

setlocale(LC\_ALL, "russian");

cout << "Программу сделал Котов.А.А. УМЛ-112\n";

cout << "Программа для поедания стада овец)\n";

int sheep;

cout << "Введите количество овец: ";

cin >> sheep;

double\* x = randomize(sheep);

double\* y = randomize(sheep);

write\_mas(x, sheep);

write\_mas(y, sheep);

int eat;

cout << "Сколько овец хочет съесть волк?: ";

cin >> eat;

for (int i = 0; i < eat; i++)

{

double avg\_x = average(x, sheep);

double avg\_y = average(y, sheep);

int max\_dst = max\_dst\_index(x, y, sheep, avg\_x, avg\_y);

cout << "Волк ест овцу в {" << x[max\_dst] << "; " << y[max\_dst] << "}\n";

x = remove\_at(x, sheep, max\_dst);

y = remove\_at(y, sheep, max\_dst);

sheep--;

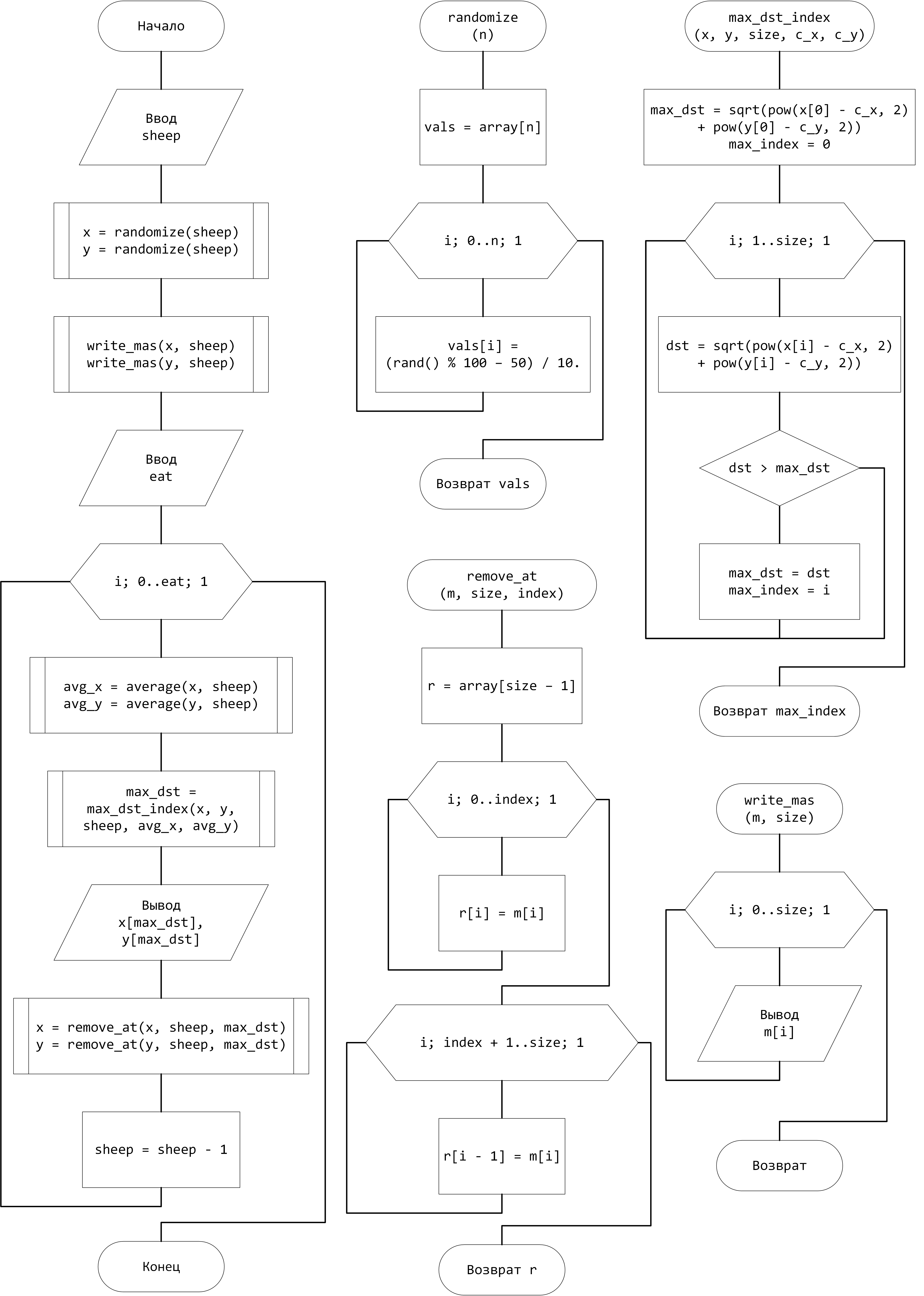
}

system("pause");

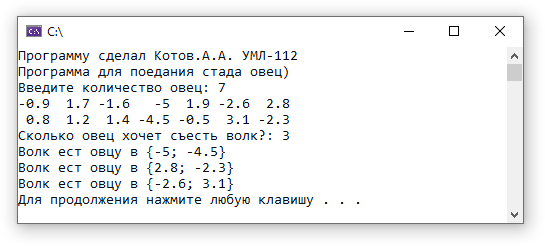
return 0;

}

**Блок-схема**



**Скриншоты работы программы:**



**Тестовые данные**

|  |  |
| --- | --- |
| Ввод | Вывод |
| 7 3 | -0.9 1.7 -1.6 -5 1.9 -2.6 2.8  0.8 1.2 1.4 -4.5 -0.5 3.1 -2.3 Волк ест овцу в {-5; -4.5}  Волк ест овцу в {2.8; -2.3}  Волк ест овцу в {-2.6; 3.1} |

**Контрольный пример** Для приведённых данных  
средний X = -0.5285, средний Y = -1.1428  
Тогда по порядку наибольшего удалению будут овцы с координатами   
{-5, -4.5}, {2.8, -2.3}, {-2.6, 3.1}.

**Вывод**: На основании того, что полученные результаты совпали с ожидаемыми, программа работает верно.