Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Основы алгоритмизации и программирования (ОАиП)

ОТЧЕТ

по разминочной работе №1

Тема работы: Нахождение расстояния между точками на координатной плоскости

Выполнил

студент: гр. 551003 Дементей В.С.

Проверила: Фадеева Е.П.

Минск 2016

Содержание

[1 Постановка задачи 3](#_Toc447385771)

[2 Описание алгоритмов 4](#_Toc447385772)

[3 Основные расчетные формулы 5](#_Toc447385773)

[4 Результаты расчетов и тестирование программы 6](#_Toc447385774)

[4.1 Тест 1 6](#_Toc447385775)

[4.2 Тест 2 6](#_Toc447385776)

[Приложение А 7](#_Toc447385777)

# Постановка задачи

На плоскости расположены 10 точек, которые заданы своими координатами. Найти на оси абсцисс точку, наибольшее расстояние от которой до выбранных точек было бы минимальным.

Ввод:

Степень точности

Вывод:

Координата точки

Расстояние

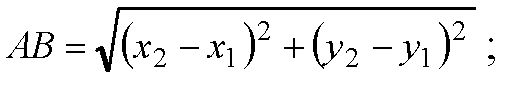
# Описание алгоритмов

Таблица – Описание алгоритмов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Наименование  алгоритма | Назначение  алгоритма | Формальные  параметры | Рекомендуемый  тип |
| 1 | Основной  алгоритм | Вызов алгоритмов ввода начальных значений, поиска границ, поиска точки и вывода результатов. |  |  |
| 2 | Border  (TestArrayC left,right) | Определяет границы прохода. | TestArrayC  Возращаемые параметры  left,right | Процедура |
| 3 | MaxDistance  (x, AnyARrayC, outCount) | Алгоритм вычисляющий максимальное расстояние от точки на оси абсцисс. | X, AnyARrayC  Возвращаемый параметр:  outCount | Процедура |
| 4 | main(Eps,  LeftB,RightB,AnyARrayC, total,  TotalDistanse) | Алгоритм для поиска точки на оси абсцисс с минимальным расстоянием до самой дальней точки. Вызывает алгоритм MaxDistance  (x, AnyARrayC, outCount) | Eps,LeftB,RightB,AnyARrayC  Возращаемые параметры:  total,  TotalDistanse | Процедура |
| 5 | enterFirstCount  (Eps) | Алгоритм ввода начальных значений. | Eps | Процедура |
| 6 | Endwrite  (total,  totaldistanse) | Алгоритм вывода конечных значений. | total,  totaldistanse | Процедура |

# Основные расчетные формулы

Расстояние между двумя точками равно квадратному корню из суммы квадратов разностей координат по каждой оси. Для того, чтобы определить расстояние между двумя точками на плоскости, надо ввести две координаты двух точек.



# Результаты расчетов и тестирование программы

## Тест 1

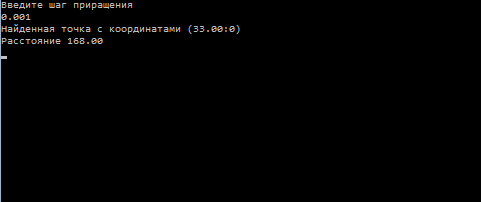
Исходные данные:

(145,33,33,122,78,126,125,114,126,138), (112,168,162,58,98,82,93,-58,-28,-62))

Eps = 0.001

Ожидаемый результат: Найденная точка с координатами (33.00:0)

Расстояние 168.00



Полученный результат:

## Тест 2

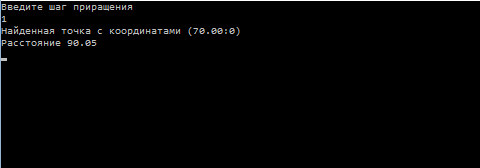
Исходные данные: ((8,15,26,42,67,33,99,115,26,11),

(65,16,19,64,73,46,81,78,32,68))

Eps = 1

Ожидаемый результат: Найденная точка с координатами (70.00:0)

Расстояние 90.05

Полученный результат:

Приложение А

(обязательное)

Исходный код программы

**Program** distanse2;

{$APPTYPE CONSOLE}

**Uses**

SysUtils,

windows;

**Type**

TArray = **array** [1..2, 1..10] **of** real;

**Const**

ARrayC: TArray = ((145,133,33,122,78,126,125,114,126,138), {X}

(112,168,162,58,98,82,93,-58,-28,-62)); {Y}

**Var**

Eps,total,TotalDistanse:real;

left,right:real;

{Алгоритм поиска правой и левой границы}

**procedure** Border(TestArrayC:TArray;**var** left:real; **var** right:real);

**var**

i:integer;

**begin**

left:=TestArrayC[1,1];

right:=TestArrayC[1,1];

**for** i:=1 **to** 10 **do**

**begin**

**if** right<TestArrayC[1,i] **then**

right:=TestArrayC[1,i];

**if** left>TestArrayC[1,i] **then**

left:=TestArrayC[1,i];

**end**;

**end**;

{Алгоритм расчета расстояния }

**procedure** MaxDistance(x:real;AnyARrayC:TArray; **Var** outCount:real);

**var**

i:integer;

Distance:real;

**begin**

outCount:=0;

**for** i:=1 **to** 10 **do**

**begin**

Distance:= sqrt(sqr(x-AnyARrayC[1,i])+sqr(0-AnyARrayC[2,i]));

**if** outCount<Distance **then**

outCount:=Distance;

**end**;

**end**;

{Алгоритм Рассчета основных значений}

**procedure** main(Eps,LeftB,RightB:real;AnyARrayC:TARray; **var** total,

TotalDistanse:real);

**var**

point,newDistance:real;

**begin**

MaxDistance(LeftB,AnyARrayC,TotalDistanse);

total:=LeftB;

point:=LeftB;

{Идем по прирощени Eps до правой границы}

**while** point <= RightB **do**

**begin**

MaxDistance(point,AnyARrayC,newDistance);

**if** newDistance<TotalDistanse **then**

**begin**

TotalDistanse:=newDistance;

total:=point;

**end**;

point:=point+Eps;

**end**;

**end**;

{Финальные вывод результатов}

**procedure** endwrite(total:real;totaldistanse:real);

**begin**

writeln('Найденная точка с координатами (', total:0:2,':0)');

writeln('Расстояние ', totaldistanse:0:2);

**end**;

{Ввод исходных значений}

**procedure** enterFirstCount(**Var** Eps:real);

**begin**

writeln('Введите шаг приращения');

readln(Eps);

**end**;

**Begin**

SetConsoleOutputCP(1251);

SetConsoleCP(1251);

enterFirstCount(Eps);

Border(ARrayC,Left,right);

main(Eps,Left,Right,ARrayC,total,totaldistanse);

endwrite(total,totaldistanse);

readln;

**End**.