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 разминке №1

Тема работы: Точки на плоскости

Выполнил

студент: гр. 251003 Панкратьев Е.С.

Проверил: Фадеева Е.П.

Минск 2023

Содержание

[1 Постановка задачи 3](#_Toc129035505)

[2 Дополненная постановка задачи 4](#_Toc129035506)

[3 Метод решения 5](#_Toc129035507)

[3.1 Описание оператора try..except..end 5](#_Toc129035508)

[3.2 Условия ввода 5](#_Toc129035509)

[3.3 Проверка введенных данных 5](#_Toc129035510)

[3.4 Краткое описание алгоритма решения задачи 5](#_Toc129035511)

[4 Описание алгоритмов решения задачи 7](#_Toc129035512)

[5 Структура данных 9](#_Toc129035513)

[6 Результаты расчетов 11](#_Toc129035514)

[Приложение А 12](#_Toc129035515)

[Приложение Б 17](#_Toc129035516)

# Постановка задачи

На плоскости расположено 10 точек, которые заданы своими координатами. Найти такую точку на оси Ох, что наибольшее расстояние от которой до заданных точек будет минимальным.

# Дополненная постановка задачи

На плоскости расположено n точек, которые заданы своими координатами. Пользователь вводит количество точек, задает их координаты и вводит точность расчета. Найти такую точку на оси Ох, что наибольшее расстояние от которой до заданных точек будет минимальным.

# Метод решения

## Описание оператора try..except..end

Оператор try..except..end имеет вид:

try  
 операторы;  
except  
  блок обработки исключений;  
end;

Выполнение блока начинается с секции try, при отсутствии исключительных ситуаций только она и выполняется. Секция except получает управление в случае возникновения исключительной ситуации. После обработки выполняются операторы, стоящие после end.

## Условия ввода

Условия ввода:

* точность вычислений может быть вещественным числом. Это значение должно быть меньше или равно 1;
* количество точек должно быть целочисленным значением и быть в диапозоне [2..10000];
* координаты точек могут быть вещественными числами.

## Проверка введенных данных

Проверка введенных данных происходит с помощью цикла с предусловием repeat..until, чтобы при вводе некорректных данных пользователь заново заполнял их. В теле цикла с помощью оператора try..except..end (описание оператора см. [главу 3.1](#_Описание_оператора_try..except..end)) проверяем соответствие с типом данным; оператором if проверяет принадлежность заданному диапазону, где это необходимо.

## Краткое описание алгоритма решения задачи

Подпрограмма A реализует алгоритм бинарного поиска, чтобы найти точку на оси x, на которой максимальное расстояние до точек минимально.

Первым шагом функция вызывает подпрограмму FindMinMax, которая находит минимальную и максимальную координаты x в массиве точек. Затем бинарный поиск начинается на интервале между этими двумя координатами.

Внутри цикла бинарного поиска сначала вычисляется середина интервала. Данная точка разбивает промежуток поиска на два. Затем с помощью подпрограммы B находятся максимальные квадраты расстояний от всех точек в массиве до центров полученных интервалов.



Рисунок 1 – Пример интервала поиска

Далее проверяется, в какой половине интервала максимальное это расстояние больше. Если максимальное расстояние меньше на левой стороне интервала, то новый интервал для поиска будет находиться между началом интервала и найденной серединой. Если максимальное расстояние меньше на правой стороне интервала, то новый интервал будет между найденной серединой и концом интервала.



Рисунок 2 – Пример интервала поиска

Бинарный поиск продолжается до тех пор, пока разница между левой и правой границей интервала не станет меньше заданной точности вычислений.

В итоге функция возвращает точку на оси x, которая соответствует минимальному максимальному расстоянию до всех точек из массива.

# Описание алгоритмов решения задачи

Таблица 1 – Описание алгоритмов решения задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  п.п. | Наименование алгоритма | Название алгоритма | Формальные  параметры | Предпола-гаемый тип реализации |
| 1 | Основной алгоритм | Ввод количества точек, ввод координат точек и ввод точности.  Вызов следующих подпрограмм:  FindPointWithMin-MaxDistance |  |  |
| 2 | FindPointWithMinMaxDistanceBin  (  Arr,  ArrLen,  CalcAccuracy,  BestPoint,  BestDist  ) | Решает задачу поиска такой точки на оси абсцисс, что расстояние между этой точкой и наиболее удаленной от нее точкой из заданного множества точек будет минимальным. Использует бинарный поиск.  Вызов следующих подпрограмм:  FindMinMax,  FindMaxSqrtDistan-ce. Лучшее расстояние сохраняет в BestPoint, лучшее расстояние в BestDist | Arr – получает от фактического параметра адрес c защитой;  ArrLen – получает от фактического параметра адрес c защитой;  CalcAccuracy – получает от фактического параметра адрес с защитой;  BestPoint, BestDist  – получает от фактического параметра адрес, возвращаемый параметр | Процедура |
| 3 | FindMinMax  (  Arr,  ArrLen,  min,  max  ) | Находит минимальное и максимальное значение координаты x в массиве точек.  Минимальное расстояние | Arr – получает от фактического параметра адрес c защитой;  ArrLen – получает от фактического параметра адрес c защитой; | Процедура |

Продолжение Таблицы 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | сохраняет в min, максимальное в max | min – получает от фактического параметра адрес,  возвращаемый параметр  max – получает от фактического параметра адрес,  возвращаемый параметр |  |
| 4 | FindMaxSqrtDistance  (  Coordx,  Arr,  ArrLen,  Result  ) | Находит максимальный квадрат расстояния между заданной точкой и точками из массива. Результат сохраняет в Result | Coordx  – получает от фактического параметра адрес c защитой;  Arr – получает от фактического параметра адрес c защитой;  ArrLen – получает от фактического параметра адрес c защитой;  Result – получает от фактического параметра адрес, возвращаемый параметр | Функция. Result – возвращае-мый функцией параметр |
| 5 | FindPointWithMinMaxDistanceLin  (  Arr,  ArrLen,  CalcAccuracy,  BestPoint,  BestDist  ) | Решает задачу поиска такой точки на оси абсцисс, что расстояние между этой точкой и наиболее удаленной от нее точкой из заданного множества точек будет минимальным.  Использует линейный поиск. | Arr – получает от фактического параметра адрес c защитой;  ArrLen – получает от фактического параметра адрес c защитой;  CalcAccuracy – получает от фактического | Процедура |

Продолжение Таблицы 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Вызов следующих подпрограмм:  FindMinMax,  FindMaxSqrtDistan-ce. Лучшую точку сохраняет в BestPoint, лучшее расстояние в BestDist | параметра адрес с защитой;  BestPoint, BestDist  – получает от фактического параметра адрес, возвращаемый параметр |  |
| 6 | RoundAcc  (  CalcAccuracy,  Result  ) | Находит количество незначащих цифр. Сохраняет результат в Result | CalcAccuracy –  получает от фактического параметра значение.  Result – получает от фактического параметра адрес, возвращаемый параметр | Функция. Result – возвращае-мый функцией параметр |

# Структура данных

Таблица 2 – Структура данных основной программы

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| MaxAmount | Integer | Максимальное количество точек |
| MinAmount | Integer | Минимальное количество точек |
| MinAccurace | Integer | Минимальная точность расчета |
| MinMaxPoint | Double | Оптимальная точка |
| Accuracy | Double | Точность расчетов |
| Amount | Integer | Количество точек |
| i | Integer | Счетчик цикла |
| ArrPoints | Array[1..10, 1..2] of Double | Массив исходных точек |
| isCorrect | Boolean | Индикатор проверки правильности ввода |

Таблица 3 – Структура данных алгоритма FindPointWithMinMaxDistanceBin

(Arr, ArrLen, CalcAccuracy, Result)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| Left | Double | Левая границы области поиска | Локальный |
| Right | Double | Правая границы области поиска | Локальный |
| Mid | Double | Середина границы | Локальный |
| Arr | Array[1..10, 1..2] of Double | Массив исходных точек | Формальный |
| ArrLen | Integer | Количество точек | Формальный |
| CalcAccuracy | double | Точность расчетов | Формальный |
| BestPoint | double | Оптимальная точка | Формальный |
| BestDist | double | Оптимальное расстояние | Формальный |

Таблица 4 – Структура данных алгоритма FindMinMax

(Arr, ArrLen, min, max)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | | Тип параметра |
| i | Integer | Счетчик цикла | | Локальный |
| Arr | Array[1..10, 1..2] of Double | Массив исходных точек | Формальный | |
| ArrLen | Integer | Количество точек | Формальный | |

Продолжение Таблицы 4

|  |  |  |  |
| --- | --- | --- | --- |
| min | Double | Минимальное значение координаты x в массиве точек | Формальный |
| max | Double | Максимальное значение координаты x в массиве точек | Формальный |

Таблица 5 – Структура данных алгоритма FindMaxSqrtDistance

(Coordx, Arr, ArrLen, Result)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| i | Integer | Счетчик циклов | Локальный |
| sqrtDist | Double | Текущий квадрат расстояния | Локальный |
| CoordX | Double | Точка на оси Ох | Формальный |
| Arr | Array[1..10, 1..2] of Double | Массив исходных точек | Формальный |
| ArrLen | Integer | Количество точек | Формальный |
| Result | Double | Максимальный квадрат расстояния между заданной точкой и точками из массива | Формальный |

Таблица 3 – Структура данных алгоритма FindPointWithMinMaxDistanceLin

(Arr, ArrLen, CalcAccuracy, Result)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| CurrPoint | Double | Текущая точка поиска | Локальный |
| LastPoint | Double | Последняя точка поиска | Локальный |
| CurrDist | Double | Лучшая точка | Локальный |
| Arr | Array[1..10, 1..2] of Double | Массив исходных точек | Формальный |
| ArrLen | Integer | Количество точек | Формальный |
| CalcAccuracy | double | Точность расчетов | Формальный |
| BestPoint | double | Оптимальная точка | Формальный |
| BestDist | double | Оптимальное расстояние | Формальный |

Таблица 4 – Структура данных алгоритма RoundAcc(CalcAccuracy, Result)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| CalcAccuracy | Double | Точность расчетов | Формальный |
| Result | Integer | Количество значащих цифр | Формальный |

# Результаты расчетов

В ходе работы были получены следующие результаты:

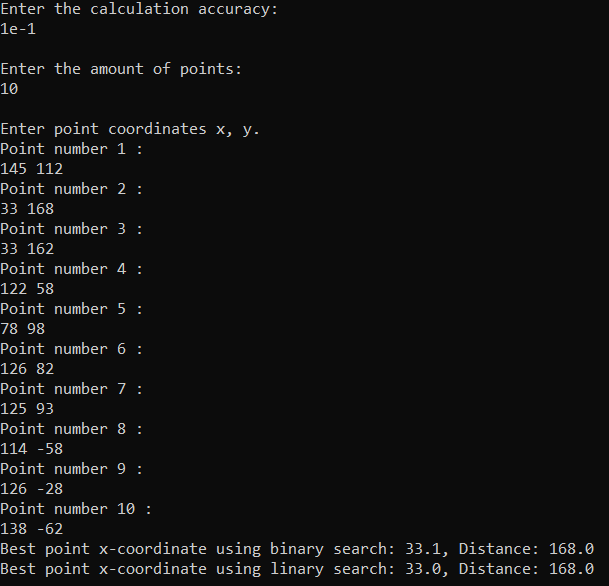


Рисунок 1 – Результаты расчетов

Приложение А

(обязательное)

Исходный код программы

program WarmUp1;

{

Given points on a plane, find a point on the x such that the

maximum

distance from that point to the given points is the minimum.

}

{$APPTYPE CONSOLE}

uses

System.SysUtils;

const

MaxAmount = 10000;

MinAmount = 2;

MinAccurace = 1;

// MaxAmount - maximum amount of points

// MinAmount - minimum amount of points

// MinAccurace - minimum accuracy

type

TNameCoord = (x, y);

TCoordsPoint = array[1..MaxAmount, TNameCoord] of double;

// TNameCoord - coordinate name

// TCoordsPoint - array of points

var

Point, Dist, Accuracy: double;

Amount, i: Integer;

ArrPoints: TCoordsPoint;

isCorrect : boolean;

// BestPoint - best point

// Accuracy - calculation accuracy

// Amount - amount of points

// i - cycle counter

// ArrPoints - array of points

// isCorrect - flag to confirm the correctness of entering numbers

// Finding the maximum sqrt(distance)

function FindMaxSqrtDistance(const Coordx: Double; const Arr:

TCoordsPoint; const ArrLen: integer): Double;

var

i: Integer;

sqrtDist: Double;

// i - cycle counter

// Dist - distance between points

begin

// Reset the result

Result := 0;

// Finding the maximum distance

for i := 1 to ArrLen do

begin

// Find the distance using the formula

sqrtDist := sqr(Arr[i, x] - Coordx) + sqr(Arr[i, y] );

// Update result if needed

if sqrtDist > Result then

Result := sqrtDist;

end;

end;

// Finds the minimum and the maximum value

procedure FindMinMax(const Arr : TCoordsPoint; const ArrLen:

integer; var min, max: double);

var

i: integer;

begin

min:= Arr[1, x];

max:= Arr[1, x];

for i := 2 to ArrLen do

begin

if Arr[i, x] > max then

max:= Arr[i, x]

else if Arr[i, x] < min then

min:= Arr[i, x];

end;

end;

// Find the best point using linear search

procedure FindPointWithMinMaxDistanceLin(const Arr:

TCoordsPoint; Const ArrLen: integer; Const

CalcAccuracy: double; var BestPoint, BestDist:

double);

var

CurrPoint, LastPoint, CurrDist: Double;

// CurrPoint - current point on the axic

// CurrPoint - last point on the axic

begin

FindMinMax(Arr, ArrLen, CurrPoint, LastPoint);

BestDist := double.MaxValue;

repeat

CurrDist:= FindMaxSqrtDistance(CurrPoint, Arr, ArrLen);

if CurrDist < BestDist then

begin

BestDist:= CurrDist;

BestPoint:= CurrPoint;

end;

CurrPoint:= CurrPoint + CalcAccuracy;

until CurrPoint > LastPoint;

BestDist:= sqrt(BestDist);

end;

// Find the best point using binary search

procedure FindPointWithMinMaxDistanceBin(const Arr:

TCoordsPoint; Const ArrLen: integer; Const

CalcAccuracy: double; var BestPoint, BestDist:

double);

var

Left, Mid, Right : Double;

// Left, Mid, Right - left, mid and right bounds of binary search

begin

// Border initialization

FindMinMax(Arr, ArrLen, Left, Right);

// Find a value until reach the given accuracy

repeat

// Finding the middle

Mid := (Left + Right) / 2;

// Narrowing down the search interval

if FindMaxSqrtDistance((Mid + Left) / 2, Arr, ArrLen) <

FindMaxSqrtDistance((Right + Mid) / 2, Arr,

ArrLen) then

Right := Mid

else

Left := Mid;

until Right - Left < CalcAccuracy;

BestPoint:= Mid;

BestDist:= sqrt(FindMaxSqrtDistance(Mid, Arr, ArrLen));

end;

function RoundAcc(CalcAccuracy: double): integer;

begin

result := 0;

while CalcAccuracy < 1 do

begin

CalcAccuracy := CalcAccuracy \* 10;

Inc(result);

end;

end;

begin

Writeln('The program find a point on the x such that the

maximum distance from that point to the given

points is the minimum.');

Writeln;

Writeln('Enter the calculation accuracy:');

// Validating the correct input data type

repeat

isCorrect:= True;

// Validate data type

Try

Readln(Accuracy);

Except

Writeln('Accuracy entered incorrectly');

isCorrect:= False;

End;

// Validate range

if isCorrect and (Accuracy > MinAccurace) then

begin

Writeln('Accuracy entered incorrectly');

isCorrect:= False;

end;

until isCorrect;

Writeln;

Writeln('Enter the amount of points:');

// Validating the correct input data type

repeat

isCorrect:= True;

// Validate data type

Try

Readln(Amount);

Except

Writeln('Amount of points entered incorrectly');

isCorrect:= False;

End;

// Validate range

if isCorrect and ((Amount < MinAmount) or (Amount >

MaxAmount)) then

begin

Writeln('Amount of points entered incorrectly');

isCorrect:= False;

end;

until isCorrect;

Writeln;

Writeln('Enter point coordinates x, y.');

// Cycle to read all the points

for i := 1 to Amount do begin

Writeln('Point number ', i,' :');

// Validating the correct input data type

repeat

isCorrect:= True;

// Validate data type

Try

Read(ArrPoints[i, x], ArrPoints[i, y]);

Except

Writeln('The entered incorrectly');

isCorrect:= False;

End;

until isCorrect;

end;

// Find the best point using binary search:

FindPointWithMinMaxDistanceBin(ArrPoints, Amount,

Accuracy, Point, Dist);

Writeln('Best point x-coordinate using binary search: ',

Point:0:1,', Distance: ', Dist:0:1);

Readln;

FindPointWithMinMaxDistanceLin(ArrPoints, Amount,

Accuracy, Point, Dist);

Writeln('Best point x-coordinate using linary search: ',

Point:0:1,', Distance: ', Dist:0:1);

Readln;

Readln;

end.

Приложение Б

(обязательное)

Тестовые наборы

**Тестовая ситуация: некорректный ввод данных**

**Тест 1**

Исходные данные: Некорректный ввод дат

Ожидаемый результат: Повторная попытка

Полученный результат:

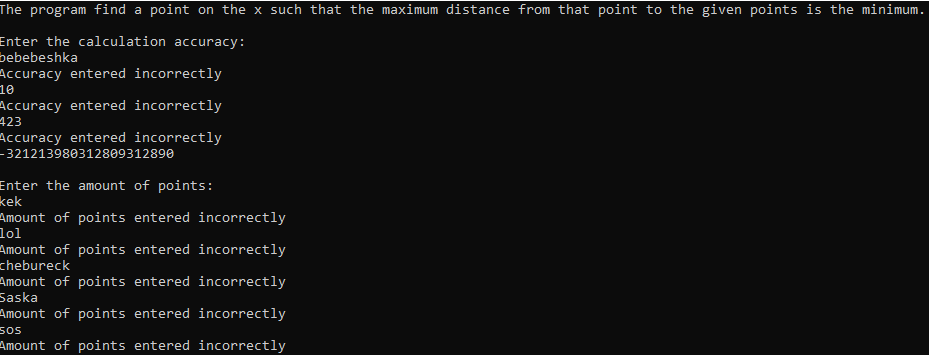


Рисунок 2 – Результаты расчетов

**Тестовая ситуация: корректный ввод данных**

**Тест 2**

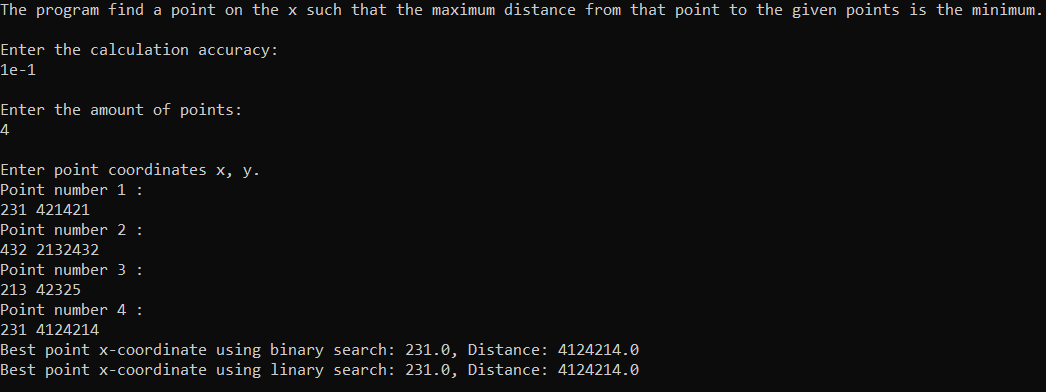


Рисунок 3 – Результаты расчетов

**Тест 3**

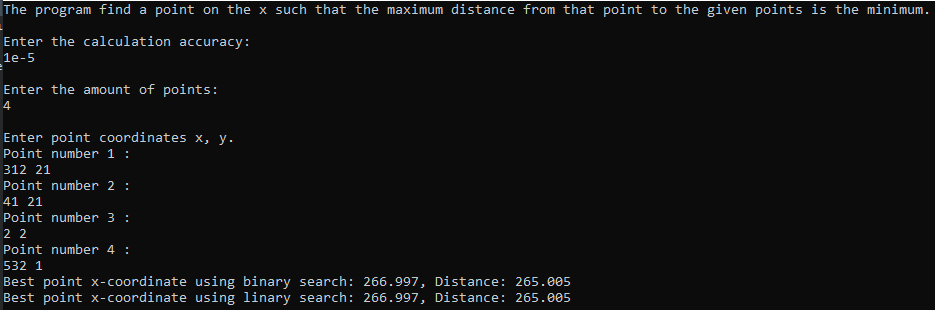


Рисунок 4 – Результаты расчетов

**Тест 6**

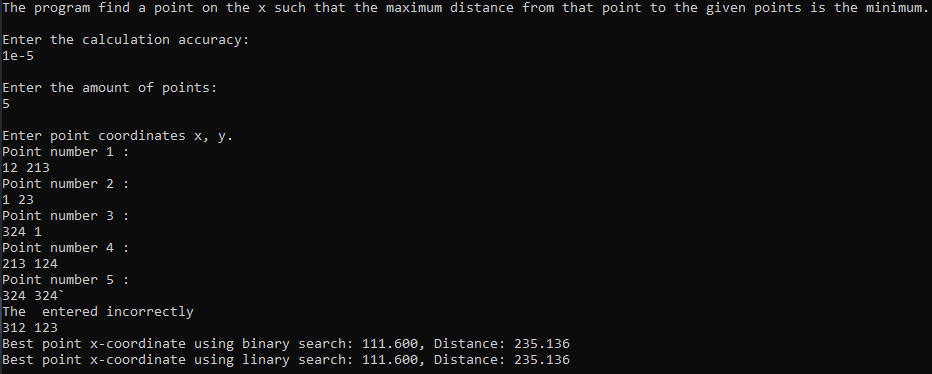


Рисунок 5 – Результаты расчетов