#include <iostream>

#include <clocale>

#include <math.h>

using namespace std;

class Isosceles\_trapezoid

{

private:

int\* x\_y1 = new int[2];

int\* x\_y2 = new int[2];

int\* x\_y3 = new int[2];

int\* x\_y4 = new int[2];

public:

Isosceles\_trapezoid(int x\_y1\_[], int x\_y2\_[], int x\_y3\_[], int x\_y4\_[])

{

x\_y1 = x\_y1\_;

x\_y2 = x\_y2\_;

x\_y3 = x\_y3\_;

x\_y4 = x\_y4\_;

}

void is\_isosceles(int x\_y1\_[], int x\_y2\_[], int x\_y3\_[], int x\_y4\_[])

{

double side1 = sqrt(pow((x\_y1[0] - x\_y2[0]), 2) + pow((x\_y1[1] - x\_y2[1]), 2));

double side2 = sqrt(pow((x\_y3[0] - x\_y4[0]), 2) + pow((x\_y3[1] - x\_y4[1]), 2));

if (side1 == side2) cout << "трапеция равнобедренная" << endl;

else cout << "трапеция не равнобедренная" << endl;

}

void lenght\_side1(int x\_y1\_[], int x\_y2\_[], int x\_y3\_[], int x\_y4\_[])

{

double side1 = sqrt(pow((x\_y1[0] - x\_y2[0]), 2) + pow((x\_y1[1] - x\_y2[1]), 2));

cout << "длина первой стороны " << side1 << endl;

}

void lenght\_side2(int x\_y1\_[], int x\_y2\_[], int x\_y3\_[], int x\_y4\_[])

{

double side2 = sqrt(pow((x\_y3[0] - x\_y4[0]), 2) + pow((x\_y3[1] - x\_y4[1]), 2));

cout << "длина второй стороны " << side2 << endl;

}

void lenght\_side3(int x\_y1\_[], int x\_y2\_[], int x\_y3\_[], int x\_y4\_[])

{

double side3 = sqrt(pow((x\_y1[0] - x\_y4[0]), 2) + pow((x\_y1[1] - x\_y4[1]), 2));

cout << "длина третей стороны " << side3 << endl;

}

void lenght\_side4(int x\_y1\_[], int x\_y2\_[], int x\_y3\_[], int x\_y4\_[])

{

double side4 = sqrt(pow((x\_y3[0] - x\_y2[0]), 2) + pow((x\_y3[1] - x\_y2[1]), 2));

cout << "длина четвёртой стороны " << side4 << endl;

}

void perimeter(int x\_y1\_[], int x\_y2\_[], int x\_y3\_[], int x\_y4\_[])

{

double side1 = sqrt(pow((x\_y1[0] - x\_y2[0]), 2) + pow((x\_y1[1] - x\_y2[1]), 2));

double side2 = sqrt(pow((x\_y3[0] - x\_y4[0]), 2) + pow((x\_y3[1] - x\_y4[1]), 2));

double side3 = sqrt(pow((x\_y1[0] - x\_y4[0]), 2) + pow((x\_y1[1] - x\_y4[1]), 2));

double side4 = sqrt(pow((x\_y3[0] - x\_y2[0]), 2) + pow((x\_y3[1] - x\_y2[1]), 2));

cout << "периметр трапеции: " << side1 + side2 + side3 + side4 << endl;

}

double square(int x\_y1\_[], int x\_y2\_[], int x\_y3\_[], int x\_y4\_[])

{

double side1 = sqrt(pow((x\_y1[0] - x\_y2[0]), 2) + pow((x\_y1[1] - x\_y2[1]), 2));

double side2 = sqrt(pow((x\_y1[0] - x\_y4[0]), 2) + pow((x\_y1[1] - x\_y4[1]), 2));

double side3 = sqrt(pow((x\_y3[0] - x\_y2[0]), 2) + pow((x\_y3[1] - x\_y2[1]), 2));

double a = acos(((x\_y1[0] - x\_y2[0]) \* (x\_y1[0] - x\_y4[0]) + (x\_y1[1] - x\_y2[1]) \* (x\_y1[1] - x\_y4[1])) / (side1 \* side2)) \* 180 / 3.1415;

double h = sin(a \* 3.1415 / 180.0) \* side1;

double s = (side2 + side3) / 2.0 \* h;

//cout << "площадь трапеции " << s << endl;

return s;

}

};

int main()

{

setlocale(LC\_CTYPE, "Rus");

int\* x\_y1 = new int[2] {-2, 0};

int\* x\_y2 = new int[2] {-1, 1};

int\* x\_y3 = new int[2] {1, 1};

int\* x\_y4 = new int[2] {2, 0};

Isosceles\_trapezoid trapezoid(x\_y1, x\_y2, x\_y3, x\_y4);

trapezoid.is\_isosceles(x\_y1, x\_y2, x\_y3, x\_y4);

trapezoid.lenght\_side1(x\_y1, x\_y2, x\_y3, x\_y4);

trapezoid.lenght\_side2(x\_y1, x\_y2, x\_y3, x\_y4);

trapezoid.lenght\_side3(x\_y1, x\_y2, x\_y3, x\_y4);

trapezoid.lenght\_side4(x\_y1, x\_y2, x\_y3, x\_y4);

trapezoid.perimeter(x\_y1, x\_y2, x\_y3, x\_y4);

cout << "площадь трапеции " << trapezoid.square(x\_y1, x\_y2, x\_y3, x\_y4);

//дано 4 трапеции

Isosceles\_trapezoid trapezoid1(

new int[2] {-4, 0},

new int[2] {-1, 1},

new int[2] {0, 1},

new int[2] {1, 0}

);

Isosceles\_trapezoid trapezoid2(

new int[2] {-1, 0},

new int[2] {0, 1},

new int[2] {1, 1},

new int[2] {2, 0}

);

Isosceles\_trapezoid trapezoid3(

new int[2] {-2, 0},

new int[2] {-1, 1},

new int[2] {0, 1},

new int[2] {2, 0}

);

Isosceles\_trapezoid trapezoid4(

new int[2] {-4, 0},

new int[2] {-1, 1},

new int[2] {0, 1},

new int[2] {1, 0}

);

double S1 = trapezoid1.square(

new int[2] {-4, 0},

new int[2] {-1, 1},

new int[2] {0, 1},

new int[2] {1, 0}

);

double S2 = trapezoid2.square(

new int[2] {-1, 0},

new int[2] {0, 1},

new int[2] {1, 1},

new int[2] {2, 0}

);

double S3 = trapezoid3.square(

new int[2] {-2, 0},

new int[2] {-1, 1},

new int[2] {0, 1},

new int[2] {2, 0}

);

double S4 = trapezoid4.square(

new int[2] {-4, 0},

new int[2] {-1, 1},

new int[2] {0, 1},

new int[2] {1, 0}

);

double\* squares = new double[4] {S1, S2, S3, S4};

double arithmetic\_mean = (S1 + S2 + S3 + S4)/4.0;

int count = 0;

for (int i = 0; i <= 3; i++)

{

if (squares[i] > arithmetic\_mean)

{

count++;

}

}

cout << "количество трапеций c полщадью больше средней: " << count << endl;

}