Bridgette Stranko

Scientific Programming

Dr. Joseph P. Allen

October 1, 2013

**Problem:**

Find the equation of the line that passes through R and is perpendicular to the line PQ. Then find the coordinates of the point of intersection of the lines PQ and the line through R that is perpendicular to PQ. The user will provide the coordinates of the three points.

The following data should be used as the baseline case:

P: (-1,0) Q: (2,3) R: (-2,4)

**Solution:**

The coordinates for P, Q, R are (a,b), (c,d) and (e,f) respectively.

To find the equation for the line that passes through points P and Q use the slope intercept formula

The formula is y = m1x+b1

Where the slope, or m1 = (d – b) / (c-a)

And the y intercept, or b1 = (b\*c-d\*a) / (c-a)

To find the equation for the perpendicular line that passes through R use the slope intercept formula where the slope from the line that passes through P and Q is inverted

The formula is y = m2x+b2

Where the slope, or m2 =-(c-a) / (d – b)

And the y intercept, or b2 = (f\*d-f\*b + c\*e-a\*e) / (d-b)

To find the point of intersection of the two lines

First substitute the y in the first equation with the solution for y in the second equation and solve for x

m2\*x+b2 =m1x+b1

xT = (b1-b2) / (m2-m1)

Second substitute the x value above in either the first equation or the second equation and solve for y

y = m1\*[(b1-b2)/(m2-m1)] + b1

The coordinate is (xT,yT)

Where xT = (b1-b2) / (m2-m1)

And yT = m1\*[(b1-b2)/(m2-m1)] + b1

And m1 = (d – b) / (c-a)

And b1 = (b\*c-d\*a) / (c-a)

And m2 =-(c-a) / (d – b)

And b2 = (f\*d-f\*b + c\*e-a\*e) / (d-b)

The base line data generates the following results:

m1 = (3-0) / (2 – (-1)) = 1 m2 = - (2 – (-1)) / (3-0) = -1

b1 = ((0\*2) – (3 \* (-1))) / (2 – (-1)) = 1 b2 = ((4\*3) – (4\*0) + (2\*(-2)) – ((-1) \* (-2) / (3 – 0) = 2

xT = (1 – 2) / (-1-1) = 1/2 or 0.5 yT = 1 \* [ (1- 2) / (-1 - 1) ] + 1 = 1 ½ or 1.5

**Algorithm:**

Declare the values as real, decimal values for

the x coordinate for point P as a, b, c, d, e, & f

the y coordinate for point P as b

the x coordinate for point Q as c

the y coordinate for point Q as d

the x coordinate for point R as e

the y coordinate for point R as f

Declare values as real, decimal values for

the slope of line PQ as m\_one,

the y intercept of line PQ as b\_one,

the slope of line R as m\_two,

the y intercept of line R as b\_two,

the x coordinate for the point of intersection as x\_coordinate,

and y coordinate for the intersection as y\_coordinate

Get the values from the user a, b, c, d, e, f

>> a b

>> c d

>> e f

if a = c & b =d

Example: P: (3,0) Q: (3,0) R: (4,9)

Then the points P & Q are the same and no solution can be provided!

Two different points are needed to form a line!

else if m\_one - m\_two = 0

No example! This is a failsafe.

Division by zero is impossible! You sunk my battleship ☹

else if c-a = 0 and c ≠ 0

Example: P: (3,1) Q: (3,5) R: (6,9)

Then the slope is undefined for line PQ

There is no y intercept for line PQ

The slope is: 0 for the line that passes through R

The y intercept is: point f for the line that passes through R

The point of intersection for these two lines is: (a,f)

else if c–a= 0

Example: P: (0,2) Q: (0,5) R: (6,7)

Then the slope is undefined for line PQ

The line PQ is a vertical line on the y axis; the y intercept is any real number!

The slope is: 0 for the line that passes through R

The y intercept is: point f for the line that passes through R

The point of intersection for these two lines is: (a,f)

else if d - b equals 0 & e ≠ 0

Example: P: (6,3) Q: (4,3) R: (5,6)

Then the slope is 0 for line PQ

The y intercept is point b for line PQ

The slope is undefined for the line that passes through R

There is no y intercept for the line that passes through R

The point of intersection for these two lines is: (e,b)

else if d - b = 0

Example: P: (6,3) Q: (4,3) R: (0,3)

Then the slope is 0 for line PQ

The y intercept is point b for line PQ

The slope is undefined for the line that passes through R

The line that passes through R is a vertical line on the y axis.

The point of intersection for these two lines is: (e,b)

Else calculate

Assign value to the following variables by calculating the following

m\_one = (d – b) / (c-a)

b\_one = (b\*c-d\*a) / (c-a)

m\_two =-(c-a) / (d – b)

b\_two = (f\*d-f\*b + c\*e-a\*e) / (d-b)

Else calculate

x\_coordinate = (b\_one-b\_two) / (m2-m1)

y\_coordinate = m\_one\*[(b\_one-b\_two)/(m\_two-m\_one)] + b\_one

The slope for line PQ is m\_one

The y intercept for line PQ is b\_one

The slope for line PQ is m\_two

The y intercept for line PQ is b\_two

The coordinates of the point of intersection of the lines P: (a, b) Q: (c,d) and the line through R (e,f) that is perpendicular to PQ are (x\_coordinate, y\_coordinate)

**Pseudocode**

Open main

Comment: Declaring the variables where the (x and y) coordinates for P are (a,b) Q are (c, d) and f are (e,f)

Comment: The slope and y intercept for PQ is m\_one, b\_one,

Comment: The slope and y intercept for R is m\_two, b\_two,

Comment: The x and y coordinates are x\_coordinate, y\_coordinate

Declare all values as real, decimal values

Real: a, b, c, d, e, f

Real: m\_one, b\_one, m\_two, b\_two, x\_coordintate, y\_coordinate

Comment: Informing the user of the purpose of this program

To screen<<"This program: (newline)" <<end line

To screen <<"(tab)\*Asks that you provide the coordinates for points "P", "Q" "R" " << end line

To screen<<"(tab) where input data is precise to tenths (newline)" << end line

To screen<<” (tab) \*Finds the slope and y intercept of the line" << end line

To screen<<"(tab) which passing through points "P" and "Q" (newline)" << end line

To screen<<"(tab) \*Finds the slope and y intercept of the line which is " << end line

To screen<<"(tab) perpendicular to the line through "P" and "Q" " << end line

To screen<<"(tab) and which passes through a third point "R"(newline)" << end line

To screen<<"(tab) \*Then provides the point of intersection of these two lines (newline)” << end line;

Comment: Informing the user how to enter the data into the program

To screen <<"Don't forget: only enter one digit pass the decimal! (newline)"

To screen <<"Enter a space between the coordinate values then press enter (newline)" << end line

Comment: requesting the values for the points PQR from the user

To screen<<"Please provide the (x,y) coordinates of the point P: (tab)";

From screen>> a>> b;

To screen <<"Please provide the (x,y) coordinates of the point Q: (tab)";

From screen >> c>>d;

To screen <<"Please provide the (x,y) coordinates of the point R: (tab)";

From screen>> e>>f;

Comment: Setting the precision of the decimal to tenths as mentioned in the instructions above

Set decimal precision to tenths

Comment: Putting out the points to the screen P: (a,b) Q: (c,d) R: (e,f)

To screen <<"(new line) P: " <<" (" <<a <<" , " << b << ") Q: "

To screen<<" (" <<c <<" , " <<d<< ") R" << " (" <<e<< " , " <<f<< ")"

To screen <<end line

To screen <<end line

Comment: The formula that will be used to find the slopes, y intercepts and points of intersection is

Comment: m\_one = (d-b) / (c-a);

Comment: b\_one = (b\*c-d\*a) / (c-a);

Comment: m\_two =-(c-a) / (d-b);

Comment: b\_two= (f\*d-f\*b + c\*e-a\*e) / (d-b);

Comment: x\_coordinate = (b\_one - b\_two) / (m\_two - m\_one);

Comment: y\_coordinate = m\_one\*(( b\_one - b\_two)/( m\_two - m\_one)) + b\_one;

Comment: Before using the formula, the program is eliminating some of the possible points that could break the formula

if a = c & b = d

To screen <<"The points P & Q are the same and no solution can be provided!"

To screen <<"Two different points are needed to form a line!"

Comment: There is a lot of division in this formula, and the program wants to avoid division by zero

Comment: The next few statements should avoid division by zero

Comment: (m\_one - m\_two) is part of a denominator and division by zero is not possible. To see if (m\_two – m\_one) = 0 we need to first make sure that neither c-a or d-b = 0 because they are in the denominators of the slopes. Then we need to calculate ((d-b) / (c-a)) +(c-a) / (d-b)) !0. Since I cannot think of a scenario when this might happen, the output to the screen will be something silly, just in case there is a scenario that I am missing.

else if c-a!= 0 & d-b != 0 & ((d-b) / (c-a)) +(c-a) / (d-b)) =0

To Screen: “Division by zero is impossible! You sunk my battleship!”

else if c- a = 0 and c != 0

Set decimal precision to hundredths;

To Screen <<"For the line which passing through points "P" and "Q"" << end line

To Screen <<"(tab) The slope is undefined" <<end line

To Screen <<"(tab) There is no y intercept" << end line

To Screen << end line

To Screen <<"For the line which is perpendicular to the line through "P" and "Q" " << end line

To Screen <<"and which passes through a third point "R" (newline)"

To Screen <<"(tab) The slope is: 0" <<end line

To Screen <<"(tab)The y intercept is: " <<f <<end line

To Screen << end line

To Screen <<"The point of intersection for these two lines is: "

To Screen <<"(" << a << " , " << f << ")"

To Screen << end line

else if c-a = 0

Set decimal precision to hundredths;

To Screen <<"For the line which passing through points "P" and "Q"" << end line

To Screen <<"(tab)The slope is undefined" << end line

To Screen <<"(tab)PQ is a vertical line on the y axis, the y intercept is any real number!"

To Screen << end line

To Screen << end line

To Screen <<"For the line which is perpendicular to the line through "P" and "Q" " << end line

To Screen <<"and which passes through a third point "R" (new line)"

To Screen <<"(tab)The slope is: 0" << end line

To Screen <<"(tab)The y intercept is: " <<f << end line

To Screen << end line

To Screen <<"The point of intersection for these two lines is: "

To Screen <<"(" << a << " , " << f << ")"

To Screen << end line

else if d - b equals 0 & e != 0

Set decimal precision to hundredths;

To Screen <<"For the line which passing through points "P" and "Q" " << end line

To Screen <<"(tab)The slope is: 0" << end line

To Screen <<"(tab) The y intercept is: " <<b << end line

To Screen << end line

To Screen <<"For the line which is perpendicular to the line through "P" and "Q" " << end line

To Screen <<"and which passes through a third point "R" (new line)"

To Screen <<"(tab)The slope is undefined" << end line

To Screen <<"(tab) There is no y intercept" << end line

To Screen << end line

To Screen <<"The point of intersection for these two lines is: "

To Screen <<"(" << e << " , " << b << ")"

To Screen << end line

else if d - b = 0

Set decimal precision to hundredths;

To Screen <<"For the line which passing through points "P" and "Q" " << end line

To Screen <<"(tab) The slope is: 0" << end line

To Screen <<"(tab) The y intercept is: " <<b << end line

To Screen << end line

To Screen <<"For the line which is perpendicular to the line through "P" and "Q" "

To Screen << end line

To Screen <<"and which passes through a third point "R" (new line)"

To Screen <<"(tab) The slope is undefined" << end line

To Screen <<"(tab) The line that passes through R is a vertical line on the y axis." << end line

To Screen <<"(tab) the y intercept is any real number!" << end line

To Screen << end line

To Screen <<"The point of intersection for these two lines is: "

To Screen <<"(" << e << " , " << b << ")"

To Screen << end line

Comment: This system will now try to find the slopes, y intercept and point of intersection using the formula.

else

m\_one = (d-b) / (c-a);

m\_two =-(c-a) / (d-b);

b\_one = (b\*c-d\*a) / (c-a);

b\_two= (f\*d-f\*b + c\*e-a\*e) / (d-b);

x\_coordinate = (b\_one - b\_two) / (m\_two - m\_one);

y\_coordinate = m\_one\*(( b\_one - b\_two)/( m\_two - m\_one)) + b\_one;

Set decimal precision to hundredths;

To Screen <<"For the line which passing through points "P" and "Q"" << end line

To Screen <<"(tab) The slope is: " <<m\_one << end line

To Screen <<"(tab) The y intercept is: " <<b\_one << end line

To Screen << end line

To Screen <<"For the line which is perpendicular to the line through "P" and "Q" "

To Screen << endl

To Screen <<"and which passes through a third point "R" (new line)"

To Screen <<"(tab) The slope is: " <<m\_two << end line

To Screen <<"(tab) The y intercept is: " <<b\_two << end line

To Screen << end line

To Screen <<"The point of intersection for these two lines is: "

To Screen <<"(" << x\_coordinate << " , " << y\_coordinate << ")"

To Screen << end line

Return 0 to main

End main

**Source Code**

#include <iostream>

using namespace std;

int main()

{

//Declaring the variables where the (x and y) coordinates for P are (a,b) Q are (c, d) and f are (e,f)

//The slope and y intercept for PQ is m\_one, b\_one,

//The slope and y intercept for R is m\_two, b\_two,

//The x and y coordinates are x\_coordinate, y\_coordinate

double a, b, c, d, e, f;

double m\_one, b\_one, m\_two, b\_two, x\_coordinate, y\_coordinate;

//Informing the user of the purpose of this program

cout<<"This program: \n" <<endl

<<"\t \*Asks that you provide the coordinates for points \"P\", \"Q\". \"R\" " << endl

<<"\t where input data is precise to tenths \n" << endl

<<"\t \*Finds the slope and y intercept of the line" << endl

<<"\t which passing through points \"P\" and \"Q\" \n" << endl

<<"\t \*Finds the slope and y intercept of the line which is " << endl

<<"\t perpendicular to the line through \"P\" and \"Q\" " << endl

<<"\t and which passes through a third point \"R\" \n" << endl

<<"\t \*Then provides the point of intersection of these two lines \n" << endl;

//Informing the user how to enter the data into the program

cout<<"Don't forget: only enter one digit pass the decimal!\n"

<<"Enter a space between the coordinate values then press enter \n" << endl

//requesting the values for the points PQR from the user

<<"Please provide the (x,y) coordinates of the point P: \t";

cin>> a>> b;

cout <<"Please provide the (x,y) coordinates of the point Q: \t";

cin>> c>>d;

cout <<"Please provide the (x,y) coordinates of the point R: \t";

cin>> e>>f;

//Setting the precision of the decimal to tenths as mentioned in the instructions above

cout.setf(ios::fixed);

cout.setf(ios::showpoint);

cout.precision(1);

//Putting out the points to the screen P: (a,b) Q: (c,d) R: (e,f)

cout <<"\nP: " <<" (" <<a <<" , " << b << ") Q: " <<" (" <<c <<" , " <<d<< ") R" << " (" <<e<< " , " <<f<< ")"

<<endl

<<endl;

/\* The formula that will be used to find the slopes, y intercepts and points of intersection is

m\_one = (d-b) / (c-a);

b\_one = (b\*c-d\*a) / (c-a);

m\_two =-(c-a) / (d-b);

b\_two= (f\*d-f\*b + c\*e-a\*e) / (d-b);

x\_coordinate = (b\_one - b\_two) / (m\_two - m\_one);

y\_coordinate = m\_one\*(( b\_one - b\_two)/( m\_two - m\_one)) + b\_one;

Before using the formula, the progam is eliminating some of the possible points that could break the formula\*/

if ((a==c) && (b==d))

{

cout <<"The points P & Q are the same and no solution can be provided!\n"

<<"Two different points are needed to form a line!" << endl;

}

//There is a lot of division in this formula, and the program wants to avoid division by zero

//The next few statements should avoid division by zero

// (m\_one - m\_two) is part of a denominator and division by zero is not possible.

//To see if (m\_two – m\_one) = 0 we need to first make sure that neither c-a or d-b = 0

//because they are in the denominators of the slopes.

//Then we need to calculate ((d-b) / (c-a)) +(c-a) / (d-b)) !0.

//Since I cannot think of a scenario when this might happen, the output to the screen will be something //silly, just in case there is a scenario that I am missing.

else if ((c-a!= 0) && (d-b != 0) && (((d-b) / (c-a)) +(c-a) / (d-b)) == 0)

{

cout<<"Division by zero is impossible! You sunk my battleship!";

}

else if ((c-a == 0) && (c!= 0))

{

cout.precision(2);

cout <<"For the line which passing through points \"P\" and \"Q\"" << endl

<<"\t The slope is undefined" <<endl

<<"\t There is no y intercept" << endl

<< endl

<<"For the line which is perpendicular to the line through \"P\" and \"Q\" " << endl

<<"and which passes through a third point \"R\" \n"

<<"\t The slope is: 0.00" <<endl

<<"\t The y intercept is: " <<f <<endl

<< endl

<<"The point of intersection for these two lines is: "

<<"(" << a << " , " << f << ")"

<< endl;

}

else if (c-a == 0)

{

cout.precision(2);

cout <<"For the line which passing through points \"P\" and \"Q\"" << endl

<<"\t The slope is undefined" <<endl

<<"\t PQ is a vertical line on the y axis. \n"

<<"The y intercept is any real number!"

<< endl

<< endl

<<"For the line which is perpendicular to the line through \"P\" and \"Q\" " << endl

<<"and which passes through a third point \"R\" \n"

<<"\t The slope is: 0.00" <<endl

<<"\t The y intercept is: " <<f <<endl

<< endl

<<"The point of intersection for these two lines is: "

<<"(" << a << " , " << f << ")"

<< endl;

}

else if ((d-b == 0) && (e!=0))

{

cout.precision(2);

cout <<"For the line which passing through points \"P\" and \"Q\"" << endl

<<"\t The slope is: 0.00" <<endl

<<"\t The y intercept is: " <<b <<endl

<< endl

<<"For the line which is perpendicular to the line through \"P\" and \"Q\" " << endl

<<"and which passes through a third point \"R\" \n"

<<"\t The slope is undefined" <<endl

<<"\t There is no y intercept" << endl

<< endl

<<"The point of intersection for these two lines is: "

<<"(" << e << " , " << b << ")"

<< endl;

}

else if (d-b == 0)

{

cout.precision(2);

cout <<"For the line which passing through points \"P\" and \"Q\"" << endl

<<"\t The slope is: 0.00" <<endl

<<"\t The y intercept is: " <<b <<endl

<< endl

<<"For the line which is perpendicular to the line through \"P\" and \"Q\" "

<< endl

<<"and which passes through a third point \"R\" \n"

<<"\t The slope is undefined" <<endl

<<"\t The line that passes through R is a vertical line on the y axis." <<endl

<<"\t The y intercept is any real number!" << endl

<< endl

<<"The point of intersection for these two lines is: "

<<"(" << e << " , " << b << ")"

<< endl;

}

//This system will now try to find the slopes, y intercept and point of intersection using the formula

else

{

m\_one = (d-b) / (c-a);

b\_one = (b\*c-d\*a) / (c-a);

m\_two =-1\*(c-a) / (d-b);

b\_two= (f\*d-f\*b + c\*e-a\*e) / (d-b);

x\_coordinate = (b\_one - b\_two) / (m\_two - m\_one);

y\_coordinate = m\_one\*(( b\_one - b\_two)/( m\_two - m\_one)) + b\_one;

cout.precision(2);

cout <<"For the line which passing through points \"P\" and \"Q\"" << endl

<<"\t The slope is: " <<m\_one <<endl

<<"\t The y intercept is: " <<b\_one <<endl

<< endl

<<"For the line which is perpendicular to the line through \"P\" and \"Q\" "

<< endl

<<"and which passes through a third point \"R\" \n"

<<"\t The slope is: " <<m\_two <<endl

<<"\t The y intercept is: " <<b\_two <<endl

<< endl

<<"The point of intersection for these two lines is: "

<<"(" << x\_coordinate << " , " << y\_coordinate << ")"

<< endl;

}

return 0;

}