

1810705 - Application Of Maths For Software

by Kristian Cole

FILE	12521_KRISTIAN_COLE_1810705_- _APPLICATION_OF_MATHS_FOR_SOFTWARE_250776_813322322.DOCX (335.48K)		
TIME SUBMITTED	12-JUN-2020 11:03AM (UTC+0100)	WORD COUNT	2002
SUBMISSION ID	130583430	CHARACTER COUNT	11090

Table of Contents

1.0	Q1 Length	2
1.0.1	Length Of vector Calculation	2
1.0.2	Length Code	2
1.0.3	Length Test	2
2.0	Q2 Normalize	2
2.0.1.	Normalize Calculation	2
2.0.2	Normalize Code	3
2.0.3	Normalize Testing	3
3.0	Q3 Distance Between Two Points	4
3.0.1	Distance Between Two Points Calculation	4
3.0.2	Distance Between Two Points Code	5
3.0.3	Distance Between Two Points Testing	5
4.0	Q4 Dot Product Of Two Vectors	7
4.0.1	Dot Product Calculation	7
4.0.2	Dot Product Code	7
4.0.3	Dot Product Testing	8
5.0	Q5 Cross Product	9
5.0.1	Cross Product Calculations	9
5.0.2	Cross Product Code	10
5.0.3	Cross Product Testing	10
6.0	Q6 Perform a rotation about the y axis transformation	12
6.0.1	Rotation Calculation	12
6.0.2	Rotation Code	12
6.0.3	Rotation Testing	13
7.0	Q7 Mean, Mode, Standard Deviation Of A List Of Integers	14
7.0.1	Mean, Mode, Standard Deviation Calculation	14
7.0.2	Mean, Mode, Standard Deviation Code	14
7.0.3	Mean, Mode, Standard Deviation Testing	15
8.0	Q8 Union, Intersection of two sets	15
8.0.1	Union, Intersection Calculation	15
8.0.2	Union, Intersection of two sets Code	15
8.0.3	Union, Intersection of two sets Testing	16
	References	16

1.0 Q1 Length

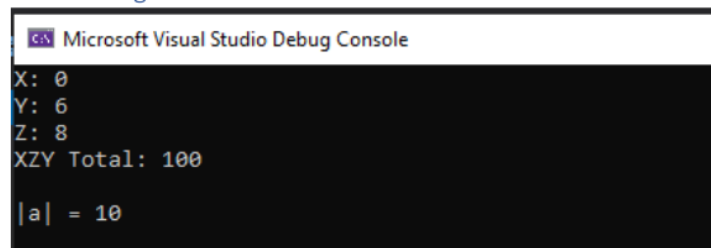
1.0.1 Length Of vector Calculation

Defined by the calculation $|\mathbf{a}| = \sqrt{x^2 + y^2 + z^2}$. $\mathbf{a}$ is the vector, $|\mathbf{a}|$ is the length of the vector, $\sqrt{x^2 + y^2 + z^2}$ is the variables squared, entirely square rooted overall [1].

1.0.2 Length Code

1. Function Vector_Length() activated from main() method.
2. Vector length code takes in three vector co ordinates (x,y,z) from user input, assigns them to a float variables (x,y,z).
3. Each entry of co-ordinates, via the standard input stream(cin), number is squared.
4. Values x,y, z are added, square rooted, the result is initialized into the length variable.

1.0.3 Length Test



```
Microsoft Visual Studio Debug Console
X: 0
Y: 6
Z: 8
XYZ Total: 100
|a| = 10
```

Test Number 1

Description: 3 dimensional length vector test

Expected Result:

X = 0
 $x^2 = 0*0*0$;
Y = 6
 $y^2 = 6*6*6 = 36$;
z = 8
 $z^2 = 8*8*8 = 64$;
 $x+y+z = 100$
 $\text{sqrt}(x+y) = 10$

Expected result 10

Actual result 10

Pass/Fail.

2.0 Q2 Normalize

2.0.1. Normalize Calculation

Normalize calculation takes the length of a vector (see vector length calculation 1.0) , each co-ordinate point (x,y,z) of the vector is divided by the length of the vector [2]. To normalize a vector divide each of the co-ordinate entries by the magnitude of the original vector

Normalisation notation $\hat{a} = \frac{a}{|a|}$, the ^ indicates the normalisation aspect of the result.

Vector $\mathbf{a}$ co-ordinates are (3,6,1) it's length is 45, normalised vector $\hat{a} = (3/45, 6/45, 1/45)$.

2.0.2 Normalize Code

1. Function `vector_Normalization()` from the `main()` function.
2. `vectType` is the choice for the user between a two or three dimensional co-ordinate.
3. `X`, `y`, `z` are the co-ordinate entries taking a float value as input.
4. `sqrX`, `sqrY`, `sqrZ` holds the squared by two values of the inputted `x`, `y`, `z` values.
5. `Length` variable holds final length of the vector, which as described in 1.0.2 is the summation of the `x`, `y`, `z` values , square rooted.
6. Output `cout` displays normalisation calculation `a = (x/length, y/length, z/length)`.

2.0.3 Normalize Testing

```
Microsoft Visual Studio Debug Console

Normalize Vector
Please Choose Vector Type

1 For XY 2 For XYZ
1
You Chose Type XY

Please Enter X:

3
X: 9
Please Enter Y:

6
Y: 36
XY Total: 45
^
a=( 3/6.7082,6 / 6.7082)
^
a=( 0.447214,0.894427 )
```

Test Number 2

Description: 2 dimensional Normalize vector test

Expected Result:

`x = 3`

`x2 = 3*3*3 = 9;`

`y = 6`

`y2 = 6*6*6= 64;`

`x+y = 45`

`sqrt(x+y) = 6.7082`

Expected result 0.44,0.89

Actual result 0.447214,0.894427

Pass/Fail.

```

Microsoft Visual Studio Debug Console

Normalize Vector
Please Choose Vector Type

1 For XY 2 For XYZ
2
You Chose Type XYZ

Please Enter X:

4
X: 16
Please Enter Y:

7
Y: 49
Please Enter z:

2
Z: 4
XYZ Total: 69
^
a=(4/8.30662,7 / 8.30662 / 2 / 8.30662)
^
a=(0.481543,0.842701 , 0.240772)

```

Test Number 3

Description: 3 dimensional Normalize vector test

Expected Result:

x = 4

$x^2 = 4 * 4 * 4 = 16;$

y = 7

$y^2 = 7 * 7 * 7 = 49;$

z = 2

$y^2 = 2 * 2 * 2 = 4;$

x+y+z = 69

sqrt(x+y+z) = 69

Expected result 0.4/0.8/2/8.3

Actual result 0.447214,0.894427,0.240772

Pass/Fail.

3.0 Q3 Distance Between Two Points

3.0.1 Distance Between Two Points Calculation

Distance between two/three co-ordinate points calculation takes two vectors, of two or three dimensions, the co-ordinates length is calculated, one vector is subtracted from the other. The length is derived from the subtraction summation between two vector points, as written below, not individually as in normalization, length calculations.

$$|\overrightarrow{AB}| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad [3]$$

For a point A = (9,7,2,1) minus Point B = (2,4,2,1)

→

$$|AB| = ((9-2)^2, (7-4)^2, (2-2)^2, (1-1)^2) = (7, 3, 0, 0) [3]$$

3.0.2 Distance Between Two Points Code

1. The function DistancebetweenTwoPoints() is activated from the main().
2. Float variables x1-3, y1-3 are initialized with zero.
3. Cin, x1-3, y1-3 are allocated co-ordinates by user input.
4. Float array distance created, initialized with x, y co-ordinates, each sequentially subtracted from each other squared.

3.0.3 Distance Between Two Points Testing

```
Microsoft Visual Studio Debug Console

Distance between two Points
Please enter X for Vector Point 1:
1
Please enter X for Vector Point 2:
5
Please enter X for Vector Point 3:
7
Please enter Y for Vector Point 1:
1
Please enter Y for Vector Point 2:
3
Please enter Y for Vector Point 3:
5
distance[0]: 0
distance[1]: 2
distance[2]: 2
distance[3]: 0
X-Y =(1-1,5-3,7-5,1-1)
X-Y =(0,2,2,0)
```

Test Number 4

Description: decimal distance between two points test

Expected Result:

X1=1

X2=5

X3=7

Y1=1

Y2=3

Y3=5

X-Y(1-1,5-3,7-5)

x-y(0,2,2,0)

Expected result x-y(0,2,2,0)

Actual result x-y(0,2,2,0)

Pass/Fail.

```
Microsoft Visual Studio Debug Console
Distance between two Points
Please enter X for Vector Point 1:
2.5
Please enter X for Vector Point 2:
1.2
Please enter X for Vector Point 3:
2.7
Please enter Y for Vector Point 1:
1.5
Please enter Y for Vector Point 2:
2.3
Please enter Y for Vector Point 3:
1.0
distance[0]: 1
distance[1]: -1.1
distance[2]: 1.7
distance[3]: 0
X-Y =(2.5-1.5,1.2-2.3,2.7-1,1-1)
X-Y =(1,-1.1,1.7,0)
```

Test Number 5

Description: Float distance between two points test

Expected Result:

X1=2.5

X2=1.2

X3=2.7

Y1=1.5

Y2=2.3

Y3=1.0

X-Y(2.5-1.5,1.2-2.3,2.7-1,1-1)

x-y(1,-1.1,1.7,0)

Expected result x-y(1,-1.1,1.7,0)

Actual result x-y(1,-1.1,1.7,0)

Pass/Fail.

4.0 Q4 Dot Product Of Two Vectors

4.0.1 Dot Product Calculation

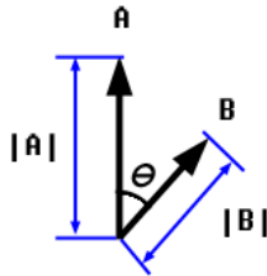


Fig. 1 Dot Product [4]

The Dot Product is the direction, length of a vector. The length of vector **a** is multiplied by vector **b** , , multiply the result by the angle derived from the length of the vectors **a**, **b**.

Can also be calculated using cosine, $\mathbf{a} \cdot \mathbf{b} = |\mathbf{a}| \times |\mathbf{b}| \times \cos(\theta)$ [4].

The usage of the angle to multiply by angle is not always needed because there is an alternate calculation where each co-ordinate entry is squared, added to each other wholly.

An example of this, $\mathbf{a} \cdot \mathbf{b} = a_x \times b_x + a_y \times b_y$

4.0.2 Dot Product Code

1. Function Dotproductoftwovectors() called from main().
2. Create array variables vectorA, vector, DotProduct, each with size of three.
3. Insert first vector into the variables x,y,z.
4. Repeat step three, insert variables x,y,z into second vector.
5. Multiply x with the corresponding y between each vector, add each of the results together.
6. Output vectorA
7. Output vectorB.
8. Output vectorA multiplied by VectorB.

4.0.3 Dot Product Testing

```
Microsoft Visual Studio Debug Console

Please Enter Vector a:

x1: 1.2
y1: 3.4
z1: 5.6

Please Enter Vector b:

x2: 2.9
y2: 4.7
z2: 1.3
a = (x1, y1, z1)  b = (x2, y2, z2) then = (x1 * x2 + y1 * y2 + z1 * z2)
a = (1.2, 3.4, 5.6)
b = (2.9, 4.7, 1.3)
ab= (1.2 * 2.9 + 3.4 * 4.7 + 5.6 * 1.3)

Dot Product: 26.74
```

Test Number 6

Description: Float Dot product test

Expected Result:

X1=1.2

Y1=3.4

Z1=5.6

X2=2.9

Y2=4.7

Z2=1.3

A=(1.2,4.7,1.3)

B=(2.9,4.7,1.3)

a.b=(1.2*2.9+3.4*4.7+5.6*1.3)

Expected result 26.74

Actual result 26.74

Pass/Fail.

```

Microsoft Visual Studio Debug Console

Please Enter Vector a:

x1: 4
y1: 2
z1: 3

Please Enter Vector b:

x2: 4
y2: 7
z2: 1
a = (x1, y1, z1)  b = (x2, y2, z2) then = (x1 * x2 + y1 * y2 + z1 * z2)
a = (4, 2, 3)
b = (4, 7, 1)
ab= (4 * 4 + 2 * 7 + 3 * 1)

Dot Product: 33

```

Test Number 7

Description: Decimal Dot product test

Expected Result:

X1=4

Y1=2

Z1=3

X2=4

Y2=7

Z2=1

A=(4,2,3)

B=(4,7,1)

a.b=(4*4+2*7+3*1)

Expected result 33

Actual result 33

Pass/Fail.

5.0 Q5 Cross Product

5.0.1 Cross Product Calculations

$$\begin{aligned}
 \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} &= a \begin{vmatrix} e & f \\ h & i \end{vmatrix} - b \begin{vmatrix} d & f \\ g & i \end{vmatrix} + c \begin{vmatrix} d & e \\ g & h \end{vmatrix} \\
 &= a(ei - fh) - b(di - fg) + c(dh - eg) \\
 &= aei - afh - bdi + bfg + cdh - ceg \\
 &= (aei + bfg + cdh) - (afh + bdi + ceg)
 \end{aligned}$$

Fig 2. Determinant matrix [5]

The cross product calculation is **axb** product that calculated another vector at a right angle **ab**. A determinant matrix calculation can be used to produce the required result [6]. Each co-ordinate

entered, from both vector **a**, **b**, entirely is added left to right into an array. The process as displayed in fig. 2 described the multiplication, subsequent subtraction.

For example vector **a** co-ordinates are (2,3,4), vector **b** is (5,6,7) , **a x b**:

$$\underline{axb} = (3 \times 7 - 4 \times 6) - (4 \times 5 - 2 \times 7) + (2 \times 6 - 3 \times 5)$$

$$\underline{axb} = -3, 6, -3$$

$$\underline{axb} = (-3, 6, -3).$$

5.0.2 Cross Product Code

1. The crossProduct() function is called from the main() .
2. Create three array variables(vectorA, vectorB, CrossProduct) with a size of three. The Crossproduct[] array will hold the matrix data to perform the determinant calculations.
3. Standard output stream (cout) requests vector co-ordinates for x, y, z with regards to vectorA. These are collected via the standard input stream(cin), co-ordinate data is allocated to array slots 0,1,2.
4. Standard output stream (cout) requests vector co-ordinates for x, y, z with regards to vectorB. These are collected via the standard input stream(cin), co-ordinate data is allocated to array slots 0,1,2.
5. Determinant matrix calculation is applied using vectorA, vectorB as input

5.0.3 Cross Product Testing

```
Microsoft Visual Studio Debug Console

Cross Product
Please Enter Vector a:

x: 4
y: 3
z: 5

Please Enter Vector b:

x: 2
y: 3
z: 1
axb = (-12 + 6 - 6)
axb = (-12,6,6)
```

Test Number 8

Description: Decimal Cross product test

Expected Result:

X1=4

Y1=3

Z1=5

X2=2

Y2=3

Z2=1

A=(4,3,5)

B=(2,3,1)

Expected result (-12+6-6)

Actual result (-12+6-6)

Pass/Fail.

Microsoft Visual Studio Debug Console

```
Cross Product
Please Enter Vector a:

x: 3.4
y: 5.4
z: 6.4

Please Enter Vector b:

x: 2.3
y: 2.1
z: 4.2
axb = (9.24 + 0.440001i - -5.28)
axb = (9.24,0.440001,-5.28)
```

Test Number 9

Description: Float Cross product test

Expected Result:

X1=3.4

Y1=5.4

Z1=6.4

X2=2.3

Y2=2.1

Z2=4.2

A=(3.4,5.4,6.4)

B=(2.3,2.1,4.2)

Expected result (9.24,0.4,-5.28)

Actual result (9.24,0.440001,-5.28)

Pass/Fail.



6.0 Q6 Perform a rotation about the y axis transformation

6.0.1 Rotation Calculation

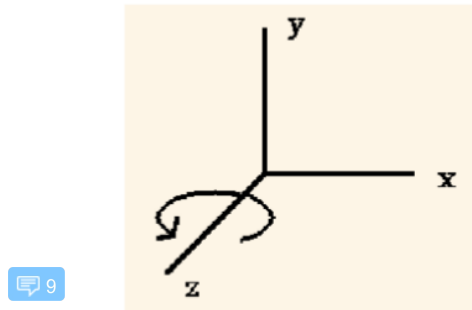


Fig 3. Rotation On Z axis [7]

The z axis rotation takes the x, y axis, performs a cos, sin function upon them, subtracts/adds the result [7].

$$\begin{aligned}y' &= y \cdot \cos q - z \cdot \sin q \\z' &= y \cdot \sin q + z \cdot \cos q \\x' &= x\end{aligned}$$

The above calculation takes the cosine theta angle of vector x, subtracts the sin right angle of vector y to reposition x, y on the z axis.

6.0.2 Rotation Code

1. Implement rotation() function via the main() .
2. Create float variables, x, y, xnew, ynew, angle, initialize them all at zero.
3. Using the standard output stream (cout) request user input on float variables x, y.
4. Using the standard input stream (cin) allocate co-ordinate float data to variables x, y.
5. The angle is calculated by dividing x by y, multiplying by 180, dividing by pi.
6. The angle, x, y is added to newly created multidimensional array variable matrix, size 2 columns, 2 rows.
7. Perform the rotation calculation described in 6.0.1.
8. Output matrix array data to the screen.
9. Calculate xnew buy subtracting matrix[0][0] by matrix[1][0]
10. Calculate ynew buy subtracting matrix[0][1] by matrix[1][1]
11. Output xnew value and pnew value to screen.

6.0.3 Rotation Testing

```
Microsoft Visual Studio Debug Console
Please enter a value for x:2
Please enter a value for y:1
Please enter an angle in degrees:3
Angle of rotation is 3 degrees
-1
0
0
0
The rotated point Pnew=(-1, 0)
```

Test Number 10

Description Decimal axis rotation test

Expected Result:

X=2

Y=1

Angle 3

Expected result (-1,0)

Actual result (-1,0)

Pass/Fail.

```
Microsoft Visual Studio Debug Console
Please enter a value for x:1.2
Please enter a value for y:3.4
Please enter an angle in degrees:23
Angle of rotation is 23 degrees
0
2
-1
0
The rotated point Pnew=(1, 2)
```

Test Number 11

Description Float axis rotation test

Expected Result:

X=1.2

Y=3.4

Angle 23

Expected result (1,2)

Actual result (1,2)

Pass/Fail.

7.0 Q7 Mean, Mode, Standard Deviation Of A List Of Integers.

7.0.1 Mean, Mode, Standard Deviation Calculation

$$\text{Mean} = \frac{\sum fx}{\sum f}$$

Fig. 4 – Mean Notation [8]

Mean calculation is the sum of all the numerical inputs, divided by the number of inputs.

Mode calculation is the frequency of one or more numbers in a data set.

$$\text{Standard deviation} = \sqrt{\frac{\sum fx^2}{\sum f} - \left(\frac{\sum fx}{\sum f}\right)^2}$$

Fig. 5 – Standard Deviation Notation [8]

Standard deviation is the summation values of mean input squared divided by how many entries into the dataset, minus the mean amount squared , the entire result is square rooted.

7.0.2 Mean, Mode, Standard Deviation Code

1. Call `standard_deviation()` from main.
2. Create float vector arrays `numData`, `mode`.
3. Create float value `mean`, integer value `check`.
4. Request numerical float data, add to `numData`.
5. Add entries to `mean`.
6. Calculate mean value, dividing by the entry amount.
7. Calculate mode by taking inputs, show frequency of the most used number.
8. Calculate standard deviation by squaring each summation mean entry input divided by input amounts. Subtract mean result squared, square root the entire result.

7.0.3 Mean, Mode, Standard Deviation Testing

```
Microsoft Visual Studio Debug Console
Please enter numbers into the dataset. Enter 0 to exit dataset entry:
23
2
33
4
6
12
0
Vector Size5
Mean: 16
Mode: 0
No Modes
4
20
56
585
1674
Total: 8.87694
```

Test Number 12

Description: Mean, Mode, Standard test

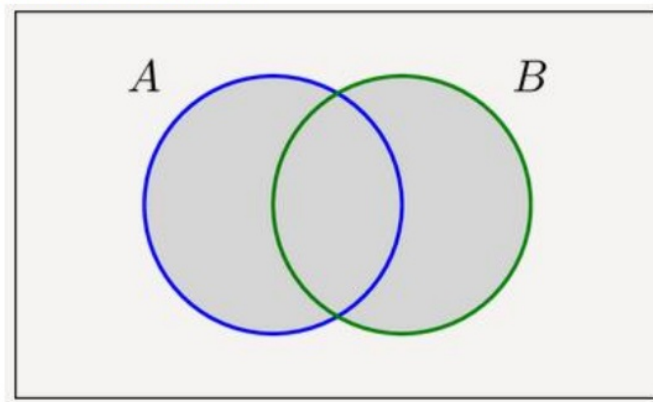
Expected result 8.87

Actual result 8.87694

Pass/Fail.

8.0 Q8 Union, Intersection of two sets

8.0.1 Union, Intersection Calculation



Two vectors (x,y) are compared to find which numbers match, this is an intersection between two vectors. The same vectors are combined, a Union. [9].

8.0.2 Union, Intersection of two sets Code

1. Call unionIntersection() function from main().
2. Create float vector array variables set1, set2, intersectionArray, unionarray
3. Create int check to use in loop.
4. Input data, enter zero to exit loop.
5. Add co-ordinate data to vector set1
6. Repeat step five, add to set2
7. Sort set1, set 2 ascending order.
8. In loop add set1 to unionArray. Compare set1 to set2 adding matches to intersectionArray.
9. Loop through intersectionArray, erase matches in set 2.

10. Add set 2 to unionArray.

8.0.3 Union, Intersection of two sets Testing

```
Microsoft Visual Studio Debug Console

Please enter numbers into the dataset 1. Enter 0 to exit dataset entry:
1
2
3
4
5
0
Please enter numbers into the dataset 2. Enter 0 to exit dataset entry:
4
5
6
7
8
0
Union 0: 1
Union 1: 2
Union 2: 3
Union 3: 4
Union 4: 5
Union 5: 6
Union 6: 7
Union 7: 8
```

Test Number 13

Description Union Intersect test

Expected Result:

$X = \{1, 2, 3, 4, 5, 6\}$

$Y = \{4, 5, 6, 7, 8\}$

Angle 23

Expected result $\{1, 2, 3, 4, 5, 6, 7, 8\}$

Actual result $\{1, 2, 3, 4, 5, 6, 7, 8\}$

Pass/Fail.

References

- [1] G. Dickers, "Vector Notes," University Of Swansea, Swansea, 2020.
- [2] G. Dickers, "Vector Recap," University Of Swansea, Swansea, 2020.
- [3] R. ehf and J. Ísak, "Mathscene - Vectors - Lesson 3," Mathscene, 2008. [Online]. Available: <http://www.rasmus.is/uk/t/F/Su58k03.htm>. [Accessed 20 04 2020].
- [4] R. Dunlop, "Dot Product," 2005.

- [5] D. Rodriguez-Clark, "Hill Cipher - Crypto Corner," Crypto Corner, 2019. [Online]. Available: <https://crypto.interactive-maths.com/hill-cipher.html>. [Accessed 21 04 2020].
- [6] Unknown, "Maths Is Fun," Maths Is Fun, 2017. [Online]. Available: <https://www.mathsisfun.com/algebra/vectors-cross-product.html>. [Accessed 15 04 2020].
- [7] Unknown, "3D Rotation," Helsinki , 1996. [Online]. Available: <https://www.cs.helsinki.fi/group/goa/mallinnus/3dtransf/3drot.html>. [Accessed 16 04 2020].
- [8] Unknown, "Calculate Mean & Standard Deviation by Formula (O Levels)," Squarespace, 19 11 2016. [Online]. Available: <https://www.tuitionkenneth.com/blog/calculate-mean-standard-deviation-formula>. [Accessed 21 04 2020].
- [9] H. Pishro-Nik, "Set Operations," Probability Course, 2020. [Online]. Available: https://www.probabilitycourse.com/chapter1/1_2_2_set_operations.php. [Accessed 17 04 2020].
- [10] G. Dickers, "Points and Vectors and Lines," University Of Swansea, Swansea, 2020.
- [11] Unknown, "Dot Product," Maths Is Fun, 2017. [Online]. Available: <https://www.mathsisfun.com/algebra/vectors-dot-product.html>. [Accessed 15 04 2020].

1810705 - Application Of Maths For Software

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

58/100

PAGE 1

PAGE 2



Comment 1

you mean $6*6$ here



Comment 2

..and $8*8$ here



Comment 3

$9+36+1=46$ not 45

PAGE 3

PAGE 4

PAGE 5



Comment 4

This is wrong - check ref - use quality texts as refs not online blogs, etc.

The previous is correct. Sum each difference squared then square root the result.

PAGE 6



Comment 5

This is not the distance between two points it is the vector from point A to point B. You need to find the length of this vector next. That is then the distance between two points.



Comment 6

Error!. you reversed the difference of y and changed the sign. Do one or the other!



Comment 7

this is correct.



Comment 8

correct!



Comment 9

rot about the y axis diagram please



Comment 10

What about std dev? calc



Comment 11

what angle?



Comment 12

intersection is $\{4,5,6\}$ so your test fails

GRADING FORM: SMU GENERIC_GJ

KRISTIAN COLE**57.50****RESEARCH-INFORMED LITERATURE**

Refs used - avoid online refs. Use established texts.

2.5**KNOWLEDGE AND UNDERSTANDING**

Some problems in your calculations. see comments on your work showing where/which operations. Most operations attempted. Rotation done but wrong axis.

13**ANALYSIS**

Some calculations need revisiting. Most attempted correctly. Testing could be more detailed.

11.5**PRACTICAL APPLICATION AND DEP.**

Code attempted for majority of operations but see earlier comments.

28

SKILLS FOR PROFESSIONAL PRAC.



Quite detailed but some problems with descriptions of operations. Structured report, Good. Refs included.

2.5