

1.10.2

RAVULA SHASHANK REDDY - EE25BTECH11047

September 13, 2025

Question

Check whether the points $(7, 10)$, $(-2, 5)$, $(3, 4)$ form an isosceles right triangle.

The condition for two sides to be perpendicular :

$$\mathbf{n}_1^T \mathbf{n}_2 = 0$$

Theoretical Solution

Given:

$$\mathbf{A} = \begin{pmatrix} 7 \\ 10 \end{pmatrix} \quad (1)$$

$$\mathbf{B} = \begin{pmatrix} -2 \\ 5 \end{pmatrix} \quad (2)$$

$$\mathbf{C} = \begin{pmatrix} 3 \\ 4 \end{pmatrix} \quad (3)$$

Side vectors:

$$\mathbf{A} - \mathbf{B} = \begin{pmatrix} 7 \\ 10 \end{pmatrix} - \begin{pmatrix} -2 \\ 5 \end{pmatrix} = \begin{pmatrix} 9 \\ 5 \end{pmatrix} \quad (4)$$

$$\mathbf{A} - \mathbf{C} = \begin{pmatrix} 7 \\ 10 \end{pmatrix} - \begin{pmatrix} 3 \\ 4 \end{pmatrix} = \begin{pmatrix} 4 \\ 6 \end{pmatrix} \quad (5)$$

$$\mathbf{B} - \mathbf{C} = \begin{pmatrix} -2 \\ 5 \end{pmatrix} - \begin{pmatrix} 3 \\ 4 \end{pmatrix} = \begin{pmatrix} -5 \\ 1 \end{pmatrix} \quad (6)$$

Theoretical Solution

Squared lengths:

$$\|\mathbf{A} - \mathbf{B}\|^2 = (\mathbf{A} - \mathbf{B})^T (\mathbf{A} - \mathbf{B}) = 9^2 + 5^2 = 106 \quad (7)$$

$$\|\mathbf{A} - \mathbf{C}\|^2 = (\mathbf{A} - \mathbf{C})^T (\mathbf{A} - \mathbf{C}) = 4^2 + 6^2 = 52 \quad (8)$$

$$\|\mathbf{B} - \mathbf{C}\|^2 = (\mathbf{B} - \mathbf{C})^T (\mathbf{B} - \mathbf{C}) = (-5)^2 + 1^2 = 26 \quad (9)$$

Since all are distinct not Isosceles.

Right angle check:

For a right angle, the dot product of two sides must be zero.

$$(\mathbf{A} - \mathbf{B})^T (\mathbf{A} - \mathbf{C}) = (9)(4) + (5)(6) = 66 \neq 0 \quad (10)$$

$$(\mathbf{A} - \mathbf{B})^T (\mathbf{B} - \mathbf{C}) = (9)(-5) + (5)(1) = -40 \neq 0 \quad (11)$$

$$(\mathbf{A} - \mathbf{C})^T (\mathbf{B} - \mathbf{C}) = (4)(-5) + (6)(1) = -14 \neq 0 \quad (12)$$

Hence, the given points forms neither an isosceles nor a right-angled triangle.

```
#include <stdio.h>
#include <math.h>

int main() {
    // Coordinates of the points
    int Ax = 7, Ay = 10;
    int Bx = -2, By = 5;
    int Cx = 3, Cy = 4;

    // Squared lengths of sides
    int AB2 = (Ax - Bx) * (Ax - Bx) + (Ay - By) * (Ay - By);
    int AC2 = (Ax - Cx) * (Ax - Cx) + (Ay - Cy) * (Ay - Cy);
    int BC2 = (Bx - Cx) * (Bx - Cx) + (By - Cy) * (By - Cy);

    // Dot products (for right angle check)
    int dot_AB_AC = (Ax - Bx) * (Ax - Cx) + (Ay - By) * (Ay - Cy)
        ;
}
```

```
int dot_AB_BC = (Ax - Bx) * (Bx - Cx) + (Ay - By) * (By - Cy)
;
int dot_AC_BC = (Ax - Cx) * (Bx - Cx) + (Ay - Cy) * (By - Cy)
;

printf(Squared side lengths:\n);
printf(AB^2 = %d\n, AB2);
printf(AC^2 = %d\n, AC2);
printf(BC^2 = %d\n, BC2);

printf(\nDot products:\n);
printf((A-B)(A-C) = %d\n, dot_AB_AC);
printf((A-B)(B-C) = %d\n, dot_AB_BC);
printf((A-C)(B-C) = %d\n, dot_AC_BC);
```

```
// Check isosceles right triangle
if ((AB2 == AC2 && dot_AB_AC == 0) ||
    (AB2 == BC2 && dot_AB_BC == 0) ||
    (AC2 == BC2 && dot_AC_BC == 0)) {
    printf("\nThe points form an ISOSCELES RIGHT triangle.\n");
} else {
    printf("\nThe points DO NOT form an isosceles right
           triangle.\n");
}

return 0;
}
```


Python Direct Code

```
import numpy as np
import matplotlib.pyplot as plt

# local imports
from libs.line.funcs import line_gen
from libs.triangle.funcs import *

# Coordinates of the points
A = np.array([7, 10]).reshape(-1, 1)
B = np.array([-2, 5]).reshape(-1, 1)
C = np.array([3, 4]).reshape(-1, 1)

# Squared lengths of sides
AB2 = np.sum((A - B) ** 2)
AC2 = np.sum((A - C) ** 2)
BC2 = np.sum((B - C) ** 2)
```

Python Direct Code

```
# Dot products
dot_AB_AC = np.dot((A - B).T, (A - C))[0, 0]
dot_AB_BC = np.dot((A - B).T, (B - C))[0, 0]
dot_AC_BC = np.dot((A - C).T, (B - C))[0, 0]

print(Squared side lengths:)
print(fAB^2 = {AB2})
print(fAC^2 = {AC2})
print(fBC^2 = {BC2})

print(\nDot products:)
print(f(A-B)(A-C) = {dot_AB_AC})
print(f(A-B)(B-C) = {dot_AB_BC})
print(f(A-C)(B-C) = {dot_AC_BC})
```

```
# Check isosceles right triangle
if ((AB2 == AC2 and dot_AB_AC == 0) or
    (AB2 == BC2 and dot_AB_BC == 0) or
    (AC2 == BC2 and dot_AC_BC == 0)):

    result = ISOSCELES RIGHT triangle
else:
    result = NOT an isosceles right triangle

print(f'\nThe points form {result}.')
```

Python Direct Code

```
# ---- Plotting ----
x_AB = line_gen(A, B)
x_BC = line_gen(B, C)
x_CA = line_gen(C, A)

plt.plot(x_AB[0, :], x_AB[1, :], 'b')
plt.plot(x_BC[0, :], x_BC[1, :], 'b')
plt.plot(x_CA[0, :], x_CA[1, :], 'b')

# Mark points
tri_coords(A, B, C, ['A(7,10)', 'B(-2,5)', 'C(3,4)'])

# Title
plt.title(fTriangle ABC: {result})
plt.gca().set_aspect('equal', adjustable='box')
plt.grid(True)
plt.show()
```

Python Shared Output

```
import ctypes
import numpy as np
import matplotlib.pyplot as plt

# Define C int type
c_int = ctypes.c_int

def main():
    # Coordinates of the points (numpy arrays)
    A = np.array([c_int(7).value, c_int(10).value])
    B = np.array([c_int(-2).value, c_int(5).value])
    C = np.array([c_int(3).value, c_int(4).value])

    # Squared lengths of sides
    AB2 = c_int(np.sum((A - B) ** 2))
    AC2 = c_int(np.sum((A - C) ** 2))
    BC2 = c_int(np.sum((B - C) ** 2))
```

Python Shared Output

```
# Dot products (for right angle check)
dot_AB_AC = c_int(np.dot(A - B, A - C))
dot_AB_BC = c_int(np.dot(A - B, B - C))
dot_AC_BC = c_int(np.dot(A - C, B - C))

print(Squared side lengths:)
print(fAB^2 = {AB2.value})
print(fAC^2 = {AC2.value})
print(fBC^2 = {BC2.value})

print(\nDot products:)
print(f(A-B)(A-C) = {dot_AB_AC.value})
print(f(A-B)(B-C) = {dot_AB_BC.value})
print(f(A-C)(B-C) = {dot_AC_BC.value})

# Check isosceles right triangle
if ((AB2.value == AC2.value and dot_AB_AC.value == 0) or
    (AB2.value == BC2.value and dot_AB_BC.value == 0) or
```

Python Shared Output

```
(AC2.value == BC2.value and dot_AC_BC.value == 0)):
    result = ISOSCELES RIGHT triangle
else:
    result = NOT an isosceles right triangle
print(f\nThe points form {result}.)

# ---- Plotting ----
fig, ax = plt.subplots()
# Draw triangle edges
triangle = np.array([A, B, C, A]) # closed loop
ax.plot(triangle[:, 0], triangle[:, 1], 'b-o', linewidth=2)

# Annotate points
ax.text(A[0]+0.2, A[1]+0.2, A(7,10), color=red)
ax.text(B[0]+0.2, B[1]+0.2, B(-2,5), color=red)
ax.text(C[0]+0.2, C[1]+0.2, C(3,4), color=red)
```

Python Shared Output

```
# Title
ax.set_title(fTriangle ABC: {result})
ax.set_aspect('equal', adjustable='box')
ax.grid(True)

plt.show()

if __name__ == '__main__':
    main()
```


