2.4.23

J.NAVYASRI- EE25BTECH11028

september 2025

Question

Do the points $(3, 2)$, $(-2, -3)$, and $(2, 3)$ form a triangle? If so, name the type of triangle formed.

Theoretical solution

Given points,

$$A = \begin{pmatrix} 3 \\ 2 \end{pmatrix}, \quad B = \begin{pmatrix} -2 \\ -3 \end{pmatrix}, \quad C = \begin{pmatrix} 2 \\ 3 \end{pmatrix} \tag{1}$$

Form the matrix:

$$M = \begin{pmatrix} 3 & 2 & 1 \\ -2 & -3 & 1 \\ 2 & 3 & 1 \end{pmatrix} \tag{2}$$

Apply row operations:

$$R_2 \leftarrow R_2 + 2R_1, \quad R_3 \leftarrow 3R_3 - 2R_1 \quad \Rightarrow \quad \begin{pmatrix} 3 & 2 & 1 \\ 4 & 1 & 3 \\ 0 & 5 & 1 \end{pmatrix} \tag{3}$$

$$R_2 \leftarrow 3R_2 - 4R_1 \quad \Rightarrow \quad \begin{pmatrix} 3 & 2 & 1 \\ 0 & -5 & 5 \\ 0 & 5 & 1 \end{pmatrix} \tag{4}$$

Theoretical solution

$$R_3 \leftarrow R_3 + R_2 \quad \Rightarrow \quad \begin{pmatrix} 3 & 2 & 1 \\ 0 & -5 & 5 \\ 0 & 0 & 6 \end{pmatrix} \tag{5}$$

Since all three rows are nonzero:

$$\text{rank}(M) = 3 \tag{6}$$

⇒ Points are not collinear, so they form a triangle.

$$\overrightarrow{AB} = B - A = \begin{pmatrix} -5 \\ -5 \end{pmatrix}, \quad \overrightarrow{AC} = C - A = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \tag{7}$$

Theoretical solution

$$\overrightarrow{AB} \cdot \overrightarrow{AC} = (-5)(-1) + (-5)(1) = 0 \tag{8}$$

$$\Rightarrow \overrightarrow{AB} \perp \overrightarrow{AC}$$

So, the triangle is right-angled at

$$A = \begin{pmatrix} 3 \\ 2 \end{pmatrix} \tag{9}$$

$$\text{The given points form a triangle (rank} = 3\text{).} \tag{10}$$

$$\text{The triangle is right-angled at } A = \begin{pmatrix} 3 \\ 2 \end{pmatrix}. \tag{11}$$

Python Code

```
 import matplotlib.pyplot as plt
# Points
A = (3, 2)
B = (-2, -3)
C = (2, 3)

# Vectors AB, AC
AB = (B[0] - A[0], B[1] - A[1])
AC = (C[0] - A[0], C[1] - A[1])

# Dot product for perpendicular check
dot = AB[0]*AC[0] + AB[1]*AC[1]

# Plot
fig, ax = plt.subplots(figsize=(6,6))
```

Python Code

```
    # Triangle edges
ax.plot([A[0], B[0]], [A[1], B[1]], 'gray')
ax.plot([B[0], C[0]], [B[1], C[1]], 'gray')
ax.plot([C[0], A[0]], [C[1], A[1]], 'gray')

# Points
ax.scatter(A[0], A[1], color='green', marker='*', s=150, zorder
    =5, label="A (3,2)")
ax.scatter(B[0], B[1], color='red', s=80, zorder=5, label="B
    (-2,-3)")
ax.scatter(C[0], C[1], color='blue', s=80, zorder=5, label="C
    (2,3)")
```

Python Code

```
    # Labels
ax.text(A[0]+0.2, A[1]+0.2, "A (3,2)", fontsize=11, fontweight="
    bold")
ax.text(B[0]-0.8, B[1]-0.4, "B (-2,-3)", fontsize=11)
ax.text(C[0]+0.2, C[1]+0.2, "C (2,3)", fontsize=11)

# Arrows AB and AC
ax.arrow(A[0], A[1], AB[0]*0.9, AB[1]*0.9,
        head_width=0.2, length_includes_head=True, color="red",
            alpha=0.8)
ax.arrow(A[0], A[1], AC[0]*0.9, AC[1]*0.9,
        head_width=0.2, length_includes_head=True, color="blue",
            alpha=0.8)
```

Python Code

```
    # Arrow labels
ax.text(A[0] + AB[0]*0.45 - 0.2, A[1] + AB[1]*0.45 - 0.2, "AB", color="red")
ax.text(A[0] + AC[0]*0.45 + 0.1, A[1] + AC[1]*0.45 + 0.1, "AC", color="blue")

# Show dot product result
msg = f"ABAC = {dot}  {'Perpendicular' if dot == 0 else 'Not perpendicular'}"
ax.text(-4.5, 4.5, msg, bbox=dict(facecolor='white', edgecolor='black'))
```

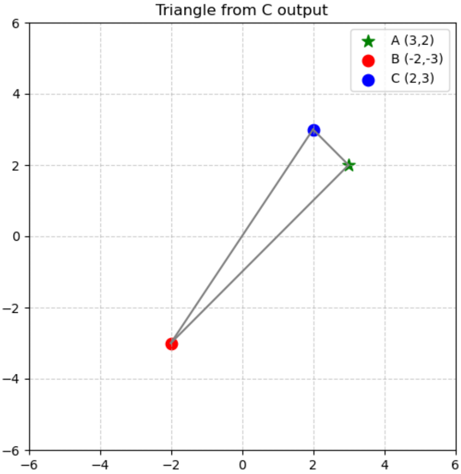
Python Code

```python
    # Axes + formatting
ax.axhline(0, color="black", linewidth=0.6)
ax.axvline(0, color="black", linewidth=0.6)
ax.set_xlim(-6, 6)
ax.set_ylim(-6, 6)
ax.set_aspect("equal", "box")
ax.grid(True, linestyle="--", alpha=0.6)
ax.set_title("Triangle ABC  check if right-angled at A")
plt.savefig("fig3.png")
plt.show()
```

C Code

```c

int main() {
    // Fixed points
    int x1 = 3, y1 = 2; // A
    int x2 = -2, y2 = -3; // B
    int x3 = 2, y3 = 3; // C


    int dot = calculateDotProduct(x1, y1, x2, y2, x3, y3);

    if(dot == 0) {
        printf("Triangle is right angled at A(3,2)\n");
    } else {
        printf("Triangle is not right angled\n");
    }

    return 0;
}
```

Plot-Using Both C and Python



Python and C Code

```python
import subprocess
import matplotlib.pyplot as plt

# Compile the C code
subprocess.run(["gcc", "triangle.c", "-o", "triangle"])

# Run the compiled program and capture output
result = subprocess.run(["./triangle"], capture_output=True, text
    =True)
coords = list(map(int, result.stdout.split()))

# Extract points
A = (coords[0], coords[1])
B = (coords[2], coords[3])
C = (coords[4], coords[5])
```

Python and C Code

```python
# ---- Plot in Python ----
fig, ax = plt.subplots(figsize=(6,6))
ax.plot([A[0],B[0]],[A[1],B[1]],'gray')
ax.plot([B[0],C[0]],[B[1],C[1]],'gray')
ax.plot([C[0],A[0]],[C[1],A[1]],'gray')

ax.scatter(*A,color='green',s=100,marker='*',label="A (3,2)")
ax.scatter(*B,color='red',s=80,label="B (-2,-3)")
ax.scatter(*C,color='blue',s=80,label="C (2,3)")
```

Python and C Code

```
ax.set_aspect("equal","box")
ax.set_xlim(-6,6)
ax.set_ylim(-6,6)
ax.grid(True,linestyle="--",alpha=0.6)
ax.set_title("Triangle from C output")
ax.legend()
plt.savefig("fig3.1.png")
plt.show()
```

Plot-Using Python

Graph representation:

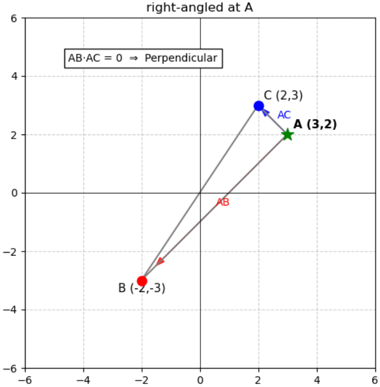


Figure: 0