2.7.24

BEERAM MADHURI - EE25BTECH11012

September 2025

Question

If the vertices of a triangle are $(1, -3)$, $(4, p)$ and $(-9, 7)$ and its area is 15 sq. units. Find the value(s) of p.

given data

let **A**, **B** and **C** be the vectors such that:

Variable	**value**
A	$\begin{pmatrix} 1 \\ -3 \end{pmatrix}$
B	$\begin{pmatrix} 4 \\ p \end{pmatrix}$
C	$\begin{pmatrix} -9 \\ 7 \end{pmatrix}$

Table: Variables used

$\text{ar}(\text{ABC}) = 15$ sq.units

Formula: Area of Triangle

$$\text{ar}(\mathbf{ABC}) = \frac{1}{2}\|(\mathbf{B} - \mathbf{A}) \times (\mathbf{C} - \mathbf{A})\|$$

finding the value of p

given ar=$(\mathbf{ABC})$ 15 sq.units

$$\text{ar}(\mathbf{ABC}) = \frac{1}{2}\|(\mathbf{B}-\mathbf{A})\times(\mathbf{C}-\mathbf{A})\| \tag{1}$$

$$= \frac{1}{2}\|\mathbf{B}\times(\mathbf{C}-\mathbf{A})-\mathbf{A}\times(\mathbf{C}-\mathbf{A})\| \tag{2}$$

$$= \frac{1}{2}\|\mathbf{B}\times\mathbf{C}-\mathbf{B}\times\mathbf{A}-\mathbf{A}\times\mathbf{C}+\mathbf{A}\times\mathbf{A}\| \tag{3}$$

$$= \frac{1}{2}\|\mathbf{B}\times(\mathbf{C}-\mathbf{A})-\mathbf{A}\times\mathbf{C}\| \tag{4}$$

Substituting the values of **A**, **B**, **C**

$$ar(\mathbf{ABC}) = 5|p+6| = 15 \tag{5}$$

$$|p+6| = 3 \tag{6}$$

$$P = -3, -9 \tag{7}$$

Hence, Value of p is -3 , -9.

Python Code

```python
import matplotlib.pyplot as plt
import numpy as np
from sympy import symbols, Eq, solve
# Given vertices
A = (1, -3)
C = (-9, 7)
# Solve for p when B = (4, p) such that area = 15
p = symbols('p', real=True)
x1, y1 = A
x2, y2 = 4, p
x3, y3 = C
```

Python Code

```python
expr = 0.5 * abs(x1*(y2 - y3) + x2*(y3 - y1) + x3*(y1 - y2))
solutions = solve(Eq(expr, 15), p)
print("Possible values of p:", solutions)

# Plot triangles for each p
fig, ax = plt.subplots(figsize=(7, 6))
```

Python Code

```
colors = ['royalblue', 'darkorange']
for sol, col in zip(solutions, colors):
    B = (4, float(sol))
    xs = [A[0], B[0], C[0], A[0]]
    ys = [A[1], B[1], C[1], A[1]]
    ax.plot(xs, ys, marker='o', color=col, label=f'p = {sol}')
    ax.text(B[0]+0.2, B[1], f'B(4,{sol.evalf():.2f})', fontsize
        =10, color=col)
```

Python Code

```python
# Mark and label points A and C
ax.text(A[0]+0.2, A[1], 'A(1,-3)', fontsize=10, color='black')
ax.text(C[0]-2, C[1], 'C(-9,7)', fontsize=10, color='black')

# Draw axes lines for reference
ax.axhline(0, color='gray', linewidth=0.8)
ax.axvline(0, color='gray', linewidth=0.8)
```

Python Code

```
ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.set_title('Triangle for given area = 15 sq.units')
ax.legend()
ax.grid(True)
plt.show()
```

C Code

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

int main() {
    double x1 = 1, y1 = -3;
    double x2 = 4, y2; // y2 = p (unknown)
    double x3 = -9, y3 = 7;
    double area = 15.0;
```

C Code

```c
    // Based on formula:
    // area = 0.5 * abs(x1*(y2 - y3) + x2*(y3 - y1) + x3*(y1 - y2))
    // We solve for y2 (p):
    // Let A = x1*(y2 - y3) + x2*(y3 - y1) + x3*(y1 - y2)
    // 2*area = |A|
    // So, A =  2*area
    double two_area = 2 * area;
```

C Code

```c
    // Express A in terms of y2:
    // A = x1*(y2 - y3) + x2*(y3 - y1) + x3*(y1 - y2)
    // = x1*y2 - x1*y3 + x2*y3 - x2*y1 + x3*y1 - x3*y2
    // Group y2 terms:
    // (x1 - x3)*y2 + (x2*y3 - x2*y1 + x3*y1 - x1*y3) = A
    double coeff_y2 = x1 - x3; // 1 - (-9) = 10
    double constant_part = x2*y3 - x2*y1 + x3*y1 - x1*y3;

    // A = coeff_y2*y2 + constant_part
```

C Code

```
    // => y2 = (A - constant_part) / coeff_y2

    // Two cases due to absolute value:
    double A1 = two_area;
    double A2 = -two_area;
    double p1 = (A1 - constant_part) / coeff_y2;
    double p2 = (A2 - constant_part) / coeff_y2;
    printf("Possible values of p are: %.2f and %.2f\n", p1, p2);
    return 0;}
```

Python and C Code

```python
import ctypes
import platform

# --- 1. Load the shared library ---
if platform.system() == "Windows":
    lib_path = "./libtriangle.dll"
else:
    lib_path = "./libtriangle.so"
try:
    lib = ctypes.CDLL(lib_path)
except OSError as e:
    print(f"Error loading library: {e}")
    print("Have you compiled triangle_solver.c?")
    exit()
```

Python and C Code

```python
# --- 2. Define the function signature ---
lib.solve_for_p.argtypes = [
    ctypes.c_double, # x1
    ctypes.c_double, # y1
    ctypes.c_double, # x2
    ctypes.c_double, # x3
    ctypes.c_double, # y3
    ctypes.c_double, # area
    ctypes.POINTER(ctypes.c_double), # p1 (output)
    ctypes.POINTER(ctypes.c_double) # p2 (output)
]
lib.solve_for_p.restype = None # void return type
```

Python and C Code

```python
# --- 3. Prepare input data and output buffers ---
# Input values from the original C code
x1, y1 = 1.0, -3.0
x2 = 4.0
x3, y3 = -9.0, 7.0
area = 15.0

# Create empty C double variables to hold the results.
# These act as output buffers.
p1_result = ctypes.c_double()
p2_result = ctypes.c_double()
```

Python and C Code

```python
# --- 4. Call the C function ---
# Pass the inputs directly and the output buffers by reference.
lib.solve_for_p(x1, y1, x2, x3, y3, area,
              ctypes.byref(p1_result),
              ctypes.byref(p2_result))
# --- 5. Print the results ---
# Access the values written by the C function using .value
print(f"Possible values of p are: {p1_result.value:.2f} and {
    p2_result.value:.2f}")
```

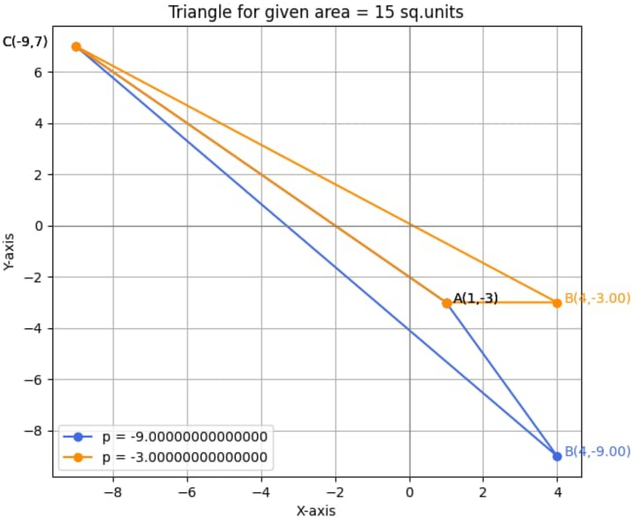
Triangle for given area = 15 sq.units
C(-9,7)
A(1,-3)
B(4,-3.00)
B(4,-9.00)
p = -9.00000000000000
p = -3.00000000000000
6
4
2
0
−2
−4
−6
−8
−8
−6
−4
−2
0
2
4
X-axis
Y-axis

Figure: Plot