

# MatGeo Presentation - Problem 1.6.14

EE25BTECH11064 - Yojit Manral

## Question

Points **A**(3, 1), **B**(12, -2) and **C**(0, 2) cannot be the vertices of a triangle.

## Solution

### Solution:

Points	Name
$\begin{pmatrix} 3 \\ 1 \end{pmatrix}$	Point <b>A</b>
$\begin{pmatrix} 12 \\ -2 \end{pmatrix}$	Point <b>B</b>
$\begin{pmatrix} 0 \\ 2 \end{pmatrix}$	Point <b>C</b>

Table: List of Points

→ Any 3 points form a triangle if the rank of co-linearity matrix is not equal to 1, in which case they become collinear. For the rank of a matrix to be 1, the number of rows with non-zero entries should be 1 in row echelon form.

## Solution

→ The co-linearity matrix is given by,

$$(\mathbf{B} - \mathbf{A} \quad \mathbf{C} - \mathbf{A})^T = \begin{pmatrix} 9 & -3 \\ -3 & 1 \end{pmatrix} \quad (0.1)$$

$$\begin{pmatrix} 9 & -3 \\ -3 & 1 \end{pmatrix} \xrightarrow{R_2 \leftrightarrow R_1 + 3R_2} \begin{pmatrix} 9 & -3 \\ 0 & 0 \end{pmatrix} \quad (0.2)$$

## Solution

→ The above matrix is in the row echelon form. Rank of the matrix in echelon form is the number of non-zero rows. Hence, rank of the above matrix is 1.

⇒ The given 3 points A, B, C are collinear. Thus, they cannot be part of a triangle.

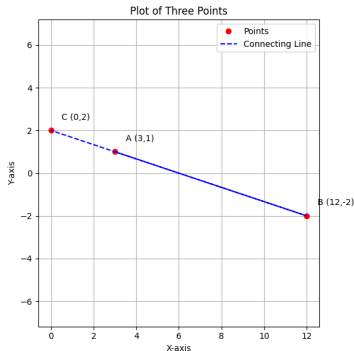


Figure: Plot of the points A, B, C

## File: points.c

```
#include <stdio.h>

int main() {
    FILE *fp;

    // -----
    // Question 1.6.14
    // -----

    fp = fopen("points.dat", "w");
    fprintf(fp, "%d,%d,%d\n", 3, 1, 0); // A
    fprintf(fp, "%d,%d,%d\n", 12, -2, 0); // B
    fprintf(fp, "%d,%d,%d\n", 0, 2, 0); // C
    fclose(fp);
    return 0;
}
```

## File: call\_c.py

```
import subprocess

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

# Run the compiled C program
result = subprocess.run(["./points"], capture_output=True, text=True)

# Print the output from the C program
print(result.stdout)
```

## File: plot.py

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

# Define the points
points = np.array([
    [3, 1], # A
    [12, -2], # B
    [0, 2] # C
])

# Extract x and y coordinates
x = points[:, 0]
y = points[:, 1]

# Plot the points
plt.figure(figsize=(6, 6))
plt.plot(x, y, 'ro', label='Points') # Red dots
plt.plot(x, y, 'b--', label='Connecting_Line') # Dashed blue line

# Annotate each point
labels = ['A(3,1)', 'B(12,-2)', 'C(0,2)']
for i in range(len(labels)):
    plt.text(x[i] + 0.5, y[i] + 0.5, labels[i], fontsize=10)

# Add plot details
plt.title('Plot of Three Points')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.grid(True)
plt.axis('equal')
plt.legend()
plt.tight_layout()
plt.show()
```