

1.2.19

Gautham-AI25BTECH11013

August 29, 2025

Question

In which quadrant or on which axis do each of the points $(-2, 4)$, $(3, -1)$, $(-1, 0)$, $(1, 2)$ and $(-3, -5)$ lie? Verify your answer by locating them on the Cartesian plane?

Theoretical Solution

If $x=0$ then the point (x, y) lies on y-axis.

If $y=0$ then the point (x, y) lies on x-axis.

If $x > 0, y > 0$ then the point (x, y) lies in 1st quadrant.

If $x < 0, y > 0$ then the point (x, y) lies in 2nd quadrant.

If $x < 0, y < 0$ then the point (x, y) lies in 3rd quadrant.

If $x > 0, y < 0$ then the point (x, y) lies in 4th quadrant.

Theoretical Solution

We can infer that $(-2, 4)$ lies in 2nd quadrant as $-2 < 0, 4 > 0$.
Similarly $(3, -1), (-1, 0), (1, 2), (-3, -5)$ lie on 4th quadrant, x-axis, 1st quadrant, 3rd quadrant respectively .

```
#include <stdio.h>

const char* check_quadrant(int x, int y) {
    if (x == 0 && y == 0) {
        return "Origin";
    } else if (x == 0) {
        return "Y-axis";
    } else if (y == 0) {
        return "X-axis";
    } else if (x > 0 && y > 0) {
        return "Quadrant I";
    } else if (x < 0 && y > 0) {
        return "Quadrant II";
    } else if (x < 0 && y < 0) {
        return "Quadrant III";
    } else {
        return "Quadrant IV";
    }
}
```

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

# Load the compiled library
quadrant_check = ctypes.CDLL('./func.so')
quadrant_check.check_quadrant.argtypes = [ctypes.c_double, ctypes.c_double]
quadrant_check.check_quadrant.restype = ctypes.c_char_p
```

```
def main():  
    # Define the points ONCE  
    points = [(-2, 4), (3, -1), (-1, 0), (1, 2), (-3, -5)]  
  
    # Check each point using the C function  
    print("Point Locations:")  
    print("=" * 40)  
  
    for i, (x, y) in enumerate(points):  
        result_bytes = quadrant_check.check_quadrant(x, y)  
        result = result_bytes.decode('utf-8')  
        print(f"Point {i+1}: ({x}, {y}) -> {result}")  
    print("=" * 40)
```

```
plt.figure(figsize=(8, 8))
plt.axhline(0, color='black') # x-axis
plt.axvline(0, color='black') # y-axis

for (x, y) in points:
    plt.scatter(x, y, s=80)
    plt.text(x+0.1, y+0.1, f"({x},{y})", fontsize=9)

plt.title("Points on Cartesian Plane (Q1.2.19)")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.grid(True)
plt.gca().set_aspect('equal', adjustable='box')
plt.savefig("/home/gauthamp/ee1030-2025/ai25btech11013/matgeo
            /1.2.19/figs/plotc.png")
plt.show()
```



```
import matplotlib.pyplot as plt

# Given points
points = [(-2, 4), (3, -1), (-1, 0), (1, 2), (-3, -5)]

# Plotting using plt only
plt.axhline(0, color='black') # x-axis
plt.axvline(0, color='black') # y-axis

for (x, y) in points:
    plt.scatter(x, y, s=80)
    plt.text(x+0.1, y+0.1, f"({x},{y})", fontsize=9)
```

```
plt.title("Points on Cartesian Plane (Q1.2.19)")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.grid(True)
plt.gca().set_aspect('equal', adjustable='box')
plt.savefig("/home/gauthamp/ee1030-2025/ai25btech11013/matgeo
            /1.2.19/figs/Figure1.png")
plt.show()
```

Plot

