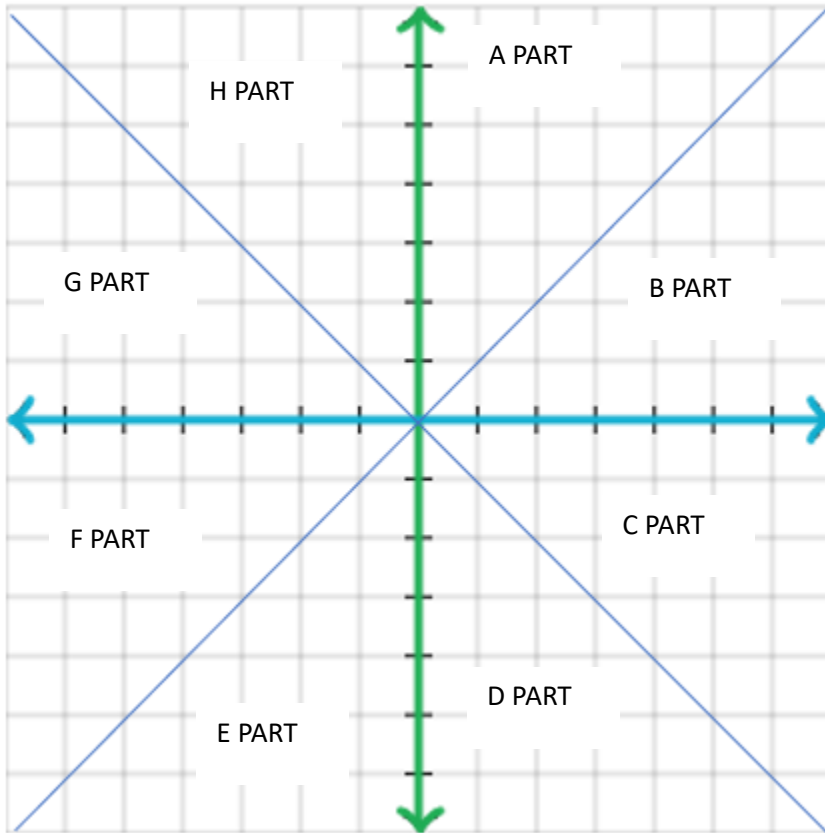


**Assignment (Programming) - 2**  
**COMP 410 – Due on 18<sup>th</sup> May 2024, 11:59 PM**

**PART A: (30)**

In the shown figure, there are 8 portions in two-dimensional space;



Write a Python program that can tell if a given point (x, y) is in any of the above part.

Use the following code to generate 10,000,000 points

```
import random
random.seed(42)
coordinate_points = [(random.randint(-5, 5), random.randint(-5, 5)) for _ in
range(10000000)]

with open("coordinate_points.txt", "w") as file:
    for point in coordinate_points:
        file.write(f"{point[0]},{point[1]}\n")
print("Coordinate points written to coordinate_points.txt")
```

This will create a text file `coordinate_points.txt`

## **Sequential Execution**

Now write your own code in another file that reads the text file and check each point and count how many points are there in each part. Use **time** module to find out the time of execution.

### **Output:**

- Your program should return the count of points in each Part.
- Print the time of execution

## **Parallel Execution**

In this part you are free to design your own approach to reduce execution time, you can use multiprocessing.

### **Output:**

- Your program should return the count of points in each Part.
- Print the time of execution

As a suggestion, use a separate function for each Part that can be called from the main program by passing the x and y coordinates of the tested point, and the function should return a binary indicator of whether the point is or is not inside the corresponding shape.

In your program, read the text file that contains xy-coordinates distributed inside the square region

$-5 < x < 5$  and  $-5 < y < 5$ .

**HINT:** You can use the Equation of line, Pythagoras' theorem

## PART B (20)

[90, 8, 80, 30, 72, 49, 79, 56, 39, 42, 93, 10, 23, 78, 7, 98, 10, 80, 26, 95, 34, 96, 83, 13, 57, 50, 49, 32, 82, 55, 69, 71, 10, 50, 31, 4, 89, 49, 99, 36, 46, 65, 46, 72, 33, 73, 49, 100, 23, 9]

Here are some numbers you need to find the Median of the list.

N = 50 (write general code don't hard code )

### Sequential Approach:

- Sort the list
- Return the middle value

### Parallel Approach:

- Sort the list in Parallel way
- Return the middle value

2<sup>nd</sup> Approach

- Explore Median of Medians Approach and Implement that approach with multiprocessing

Use time module to find the execution time of each module and compare all three approach in your report.

## Moodle Submission

Part A Report:

1-2 page Report in which you explain your approach , how you detected different regions, how you distribute the tasks, the Speedup by using Amdahl's law.

Part B Report:

Comparison of all three approaches.

- **Make one combined report of part A and Part B and submit a PDF version.**
- Submit all the code files from both parts.

**Don't upload text file of coordinate points.**

**Late Submission Policy ( 10% deduction for one day late, 25% for 2 days late, 50% for 3 days late)**