Hackathon 2024

==

You are tasked with implementing a coordinate system program in a programming language of your choice. The program should allow users to perform basic operations on points in a 2D plane. The main functionalities of the program should include:

- Point Creation: Users should be able to create points by providing x and y coordinates.
- Distance Calculation: Implement a function to calculate the Euclidean distance between two points in the coordinate system.
- Quadrant Identification: Develop a feature that identifies the quadrant in which a given point resides (1st, 2nd, 3rd, or 4th quadrant).
- Translation: Allow users to translate a point by a specified distance along the x and y axes.
- Reflection: Implement a function that reflects a point across the x or y-axis.
- Rotation: Enable users to rotate a point around the origin by a specified angle.

Ensure to include appropriate error handling and user-friendly prompts for each operation. You will have to design several API endpoints which will be tested against multiple input data. Make sure to follow specifications for each endpoint as outlined in the following sections.

The scoring for this assessment is designed to be flexible and reflective of different aspects of your solution. It allows for partial scoring, meaning that even if your solution only achieves certain criteria, you can still earn a portion of the total score. For instance, if your solution successfully produces the correct status code, you will be awarded 10% of the assigned score. The remaining 90% of the score is contingent upon the overall alignment of your solution with the response generated by our reference solution.

Problem 1:

- Endpoint: http://localhost:5000/p1
- Method: PUT
- request_body:

```
{
    "x": 12,
    "y": 13
}
```

Design an endpoint that takes two integers x and y as query param and persist the data as a two dimensional coordinate point that can be used throughout the lifetime of the application.

Upon success, echo the received parameters with http status code 201, if there is no such records are available, else status code will be 200.

- response code:

```
{
    "added": {
        "x": 12,
        "y": 13
    }
}
```

Problem 2:

- Endpoint: http://localhost:5000/p2
- Method: GET

Design an endpoint that can read the coodrinates stored in the database and calculate the center of gravity. If no point is found in the database, print the origin. Display the coordinates as integers by rounding down to the nearest whole number.

Upon success, echo the center of gravity in the following format with HTTP status code 200: response code:

```
{
    "avg": {
        "x": 120,
        "y": 200
    }
}
```

Testing

Launch the application with an empty database. Then run the program with postman or CURL.