Leetcode May Challenge DAY: 08

1. Python

```
class Solution:
    def checkStraightLine(self, coordinates: List[List[int]]) -> bool:
        x_1, y_1 = coordinates[0]
        x_2, y_2 = coordinates[1]
        x_slope = x_2 - x_1
        y_slope = y_2 - y_1
        for x, y in coordinates[2:]:
            x_s = x - x_1
            y_s = y - y_1
            if x_s * y_slope != y_s * x_slope:
                return False
        return True
```

2. C++

```
class Solution {
public:
    bool checkStraightLine(vector<vector<int>>& coordinates) {
        if(coordinates.size() <= 2) return true;
        float initSlope =
slope(coordinates[0][0],coordinates[0][1],coordinates[1][0],coordinates[1][1]);
        bool ans = true;
        for(int i = 2; i < coordinates.size(); i++) {
            if(slope(coordinates[i-1][0],coordinates[i - 1][1],
coordinates[i][0],coordinates[i][1]) != initSlope) {
                ans = false;
            }
        }
        return ans;
    }

    float slope(float x1, float y1, float x2, float y2) {
        return (y2 - y1) / (x2 - x1);
    }
};
```

3. JAVA

```
class Solution {
    public boolean checkStraightLine(int[][] cd) {
        if(cd.length==2)
            return true;
        double d=findSlope(cd[0][0],cd[0][1],cd[1][0],cd[1][1]);
        for(int i=2;i<cd.length;i++)
            if(findSlope(cd[0][0],cd[0][1],cd[i][0],cd[i][1])!=d)
                return false;
        return true;
    }
    public double findSlope(int x1, int y1, int x2, int y2){
        return (double)((double)(y2-y1)/(double)(x2-x1));
    }
}
```