

Follow me for more interesting programming questions and RAW Code visit my GitHub profile:  
<https://www.github.com/shubhamthrills>   <https://www.linkedin.com/in/shubhamsagar>

Subscribe our YouTube Channel for more videos: <https://www.youtube.com/channel/UCJLMu9mayAibQST1eWwq0cg>

## **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**

Follow me for more interesting programming questions and RAW Code visit my GitHub profile:  
<https://www.github.com/shubhamthrills>    <https://www.linkedin.com/in/shubhamsagar>

Subscribe our YouTube Channel for more videos: <https://www.youtube.com/channel/UCjLMu9mayAibQST1eWwq0cg>

## 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);**

**}**

**};**

Follow me for more interesting programming questions and RAW Code visit my GitHub profile:  
<https://www.github.com/shubhamthrills>   <https://www.linkedin.com/in/shubhamsagar>

Subscribe our YouTube Channel for more videos: <https://www.youtube.com/channel/UCjLMu9mayAibQST1eWwq0cg>

### 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));  
    }  
}
```