Judge Route Circle

Initially, there is a Robot at position (0, 0). Given a sequence of its moves, judge if this robot makes a circle, which means it moves back to **the original place**.

The move sequence is represented by a string. And each move is represent by a character. The valid robot moves are `R` (Right), `L` (Left), `U` (Up) and `D` (down). The output should be true or false representing whether the robot makes a circle.

Example 1:

```
Input: "UD"
Output: true
```

Example 2:

```
Input: "LL"
Output: false
```

Solution 1

C++

```cpp
class Solution {
public:
    bool judgeCircle(string moves) {
        int v = 0;
        int h = 0;
        for (char ch : moves) {
            switch (ch) {
                case 'U' : v++; break;
                case 'D' : v--; break;
                case 'R' : h++; break;
                case 'L' : h--; break;
            }
        }
        return v == 0 && h == 0;
    }
};
```

Java

```java
public class Solution {
    public boolean judgeCircle(String moves) {
        int x = 0;
        int y = 0;
        for (char ch : moves.toCharArray()) {
            if (ch == 'U') y++;
            if (ch == 'D') y--;
            if (ch == 'R') x++;
            if (ch == 'L') x--;
        }
        return x == 0 && y == 0;
    }
}
```

written by alexander original link here

Solution 2

```
def judgeCircle(self, moves):
    c = collections.Counter(moves)
    return c['L'] == c['R'] and c['U'] == c['D']
```

written by TianQ original link here

Solution 3

```
def judgeCircle(self, moves):
    return moves.count('L') == moves.count('R') and moves.count('U') == moves.count(
'D')
```

written by aditya74 original link here