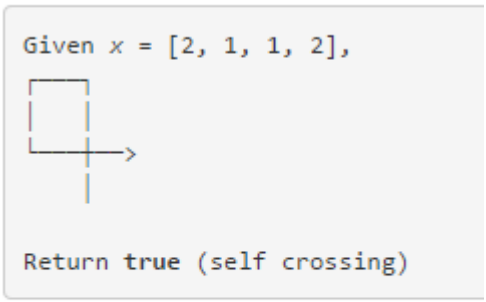
335. Self Crossing

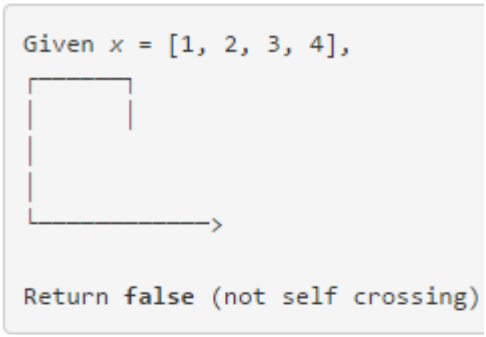
You are given an array *x* of n positive numbers. You start at point (0,0) and moves x[0] metres to the north, then x[1] metres to the west, x[2] metres to the south,x[3] metres to the east and so on. In other words, after each move your direction changes counter-clockwise.

Write a one-pass algorithm with O(1) extra space to determine, if your path crosses itself, or not.

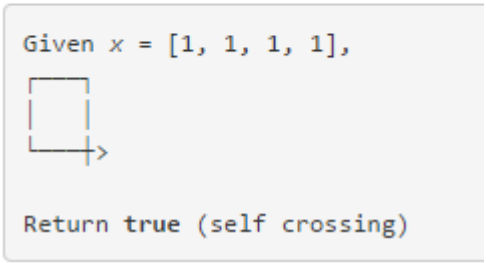
Example 1:



Example 2:



Example 3:




```
public boolean isSelfCrossing (int[] x)
{

    if (x.length < 4)
        return false;

    for (int i = 3 ; i < x.length ; i++) {
        if (isCrossing_2(x, i)) {
            return true;
```

```
        }

        if (i > 4 && isCrossing_4(x, i))
            return true;

        if (i == 4 && x[i-1] == x[i-3] && x[i] >= x[i-2] - x[i-4])
            return true;
    }

    return false;
}

public boolean isCrossing_2(int[] x, int i) {
    if (x[i-1] <= x[i-3] && x[i] >= x[i-2])
        return true;

    return false;
}


public boolean isCrossing_4(int[] x, int i) {
    if (x[i-1] <= x[i-3] && x[i-1] >= x[i-3] - x[i-5] && x[i] >=  x[i-2]
- x[i-4] && x[i-2] >= x[i-4])
        return true;

    return false;
}
```