Compute the Area of a Polygon

by PRASHANTB1984

Problem Submissions Leaderboard Discussions

You are given the cartesian coordinates of a set of points in a $2D$ plane. When traversed sequentially, these points form a Polygon, P, which is not self-intersecting in nature. Can you compute the area of polygon P?

Input Format

The first line contains an integer, N, denoting the number of points.
The N subsequent lines each contain 2 space-separated integers denoting the respective x and y coordinates of a point.

Constraints

- No 2 points are *coincident*, and polygon P is obtained by traversing the points in a counter-clockwise direction.
- $4 \leq N \leq 1000$
- $0 \leq x, y \leq 1000$

Output Format

For each test case, print the area of P (correct to a scale of one decimal place).

Note: Do not add any leading/trailing spaces or units; it is assumed that your result is in square units.

Sample Input

```
4
0 0
0 1
1 1
1 0
```

Sample Output

```
1
```

Explanation

The given polygon is a square, and each of its sides are 1 unit in length.
$area(P) = length \times width = 1 \times 1 = 1$, so we print 1 on a new line.

Submissions: 934
Max Score: 20
Difficulty: Easy

Rate This Challenge:

More

If you cannot find your favorite language here, please check out the algorithm domain. We support over 40 popular languages.

Current Buffer (saved locally, editable) Racket

```
#lang racket
; Enter your code here. Read input from STDIN. Print output to STDOUT

(define (polygon-area-iter points n sum)
  (cond ((= n (length points)) (abs (if (= (remainder sum 2) 0) (/ sum 2) (/ sum 2.0))))
        (else (polygon-area-iter
               points
               (+ n 1)
               (+ sum (-
                       (* (car (list-ref points n)) (cdr (list-ref points (remainder (+ n 1) (length points)))))
                       (* (car (list-ref points (remainder (+ n 1) (length points)))) (cdr (list-ref points n)))))))))

(define (polygon-area points) (polygon-area-iter points 0 0))

(define (read-points n)
  (cond ((= n 0) '())
        (else (cons (cons (read) (read)) (read-points (- n 1))))))

(displayln (polygon-area (read-points (read))))
```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code