Compute the Area of a Polygon ★

Points: 553.1800000000001 Rank: 1150

Problem Submissions Leaderboard

RATE THIS CHALLENGE

★ ★ ★ ★ ★

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.

Change Theme Language Haskell

```
1  import Data.List
2  import Control.Monad
3
4  calcArea :: [(Double, Double)] -> Double
5  calcArea ps = let (psx, psy) = unzip ps
6                    a = ((last psx * head psy) +) . sum $ zipWith (*) (init psx) (tail psy)
```

```
                    b = ((head psx * last psy) +) . sum $ zipWith (*) (tail psx) (init psy)
                in 0.5 * abs (a-b)


main = do
    pc <- (read::String->Double) `fmap` getLine
    ps <- forM [1..pc] $ const $ do
        a:b:_ <- (fmap (read::String->Double) . words) `fmap` getLine
        return (a,b)
    print $ calcArea ps


```

Line: 18 Col: 1

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