Consider two points, and . We consider the inversion or point reflection, , of point across point to be a rotation of point around .

Given sets of points and , find for each pair of points and print two space-separated integers denoting the respective values of and on a new line.

Function Description

Complete the findPoint function in the editor below.

findPoint has the following parameters:

int px, py, qx, qy: x and y coordinates for points and

Returns

Int[2]: x and y coordinates of the reflected point

Input Format

The first line contains an integer, , denoting the number of sets of points.

Each of the subsequent lines contains four space-separated integers that describe the respective values of , , , and defining points and .

Constraints

Sample Input

2

0 0 1 1

1 1 2 2

Sample Output

2 2

3 3

Explanation

The graphs below depict points , , and for the points given as Sample Input:

Find-point-0011.png

Find-point-1122.png

Submissions: 185

Max Score: 10

Difficulty: Easy

Rate This Challenge:

More

Return[2\*qx-px,2\*qy-py]

1

#!/bin/python3

2

3

Import math

4

Import os

5

Import random

6

Import re

7

Import sys

8

9

#

10

# Complete the ‘findPoint’ function below.

11

#

12

# The function is expected to return an INTEGER\_ARRAY.

13

# The function accepts following parameters:

14

# 1. INTEGER px

15

# 2. INTEGER py

16

# 3. INTEGER qx

17

# 4. INTEGER qy

18

#

19

20

Def findPoint(px, py, qx, qy):

21

# Write your code here

22

Return[2\*qx-px,2\*qy-py]

23

24

If \_\_name\_\_ == ‘\_\_main\_\_’:

25

Fptr = open(os.environ[‘OUTPUT\_PATH’], ‘w’)

26

27

N = int(input().strip())

28

29

For n\_itr in range(n):

30

First\_multiple\_input = input().rstrip().split()

31

32

Px = int(first\_multiple\_input[0])

33

34

Py = int(first\_multiple\_input[1])

35

36

Qx = int(first\_multiple\_input[2])

37

38

Qy = int(first\_multiple\_input[3])

39

40

Result = findPoint(px, py, qx, qy)

41

42

Fptr.write(‘ ‘.join(map(str, result)))

43

Fptr.write(‘\n’)

44

45

Fptr.close()

46

Line: 22 Col: 28

Run Code Submit CodeUpload Code as File

Test against custom input

Testcase 0

Congratulations, you passed the sample test case.

Click the Submit Code button to run your code against all the test cases.

Input (stdin)

2

0 0 1 1

1 1 2 2

Your Output (stdout)

2 2

3 3

Expected Output

2 2

3 3