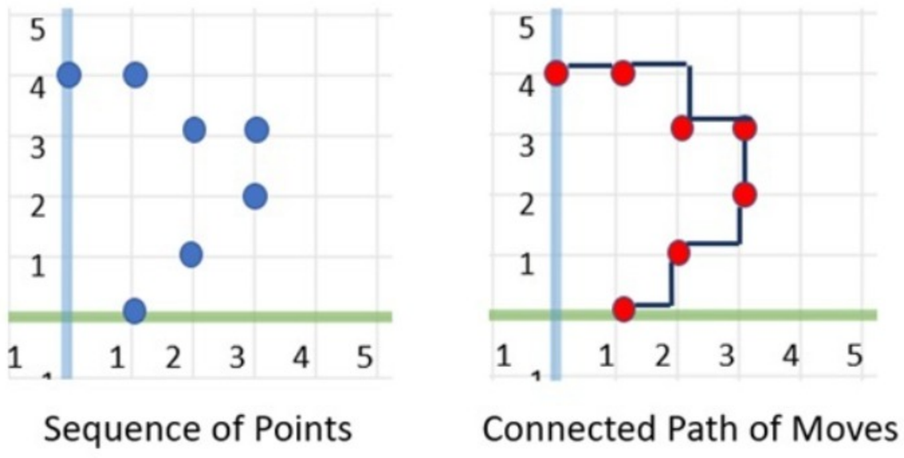
ROBOT Navigation 1

At the APEC Micromouse contest, participants need to help navigate a mouse robot in a 2D maze.

The 2D maze can be represented using the (x, y) coordinate system with origin (0,0).

The robot does not move in the diagonal direction, it only moves forward or backward, and left or right. As an example, the following image represents a path that passes through a sequence of points. It shows the path the robot navigates through the maze.



You need to compute the distance of a path that the robot navigates. The path length is given by the Manhattan distance. The Manhattan distance between two points (x1, y1) and (x2, y2) is |x1-x2| + |y1-y2|. That is, it is the sum of the absolute values of the differences between both sets of coordinates.

Input Format

The program reads integer n representing the number of points in the path, followed by n lines. Each line contains the x and y values.

Constraints

n > 0 0 <= x <= 1000000 0 <= y <= 1000000

Output Format

A single integer representing the total path length. For any invalid input the program prints -1

Sample Input 0

```
8
1 0
2 1
3 2
3 3
2 3
1 4
0 4
```

Sample Output 0

```
9
```

Explanation 0

For the given sample input, the length will be: |e1| = |1-2| + |0-1| = 1 + 1 = 2 |e2| = |2-3| + |1-2| = 1+1 = 2 |e3| = |3-3| + |2-3| = 0 + 1 = 1 |e4| = |3-2| + |3-3| = 1 + 0 = 1 |e5| = |2-1| + |3-4| = 1 + 1 = 2 |e6| = |1-0| + |4-4| = 1 + 0 = 1 Length = |e1| + |e2| + |e3| + |e4| + |e5| + |e6| = 2 + 2 + 1 + 1 + 2 + 1 = 9

Java 7

```
1  import java.io.*;
2  import java.util.*;
3  import java.text.*;
4  import java.math.*;
5  import java.util.regex.*;
6
7  public class Solution {
8
9      public static void main(String[] args) {
10         /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
11     }
12 }
```

Line: 1 Col: 1

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