# Problem D
# Texas Summers

**Problem ID:** texassumm
**CPU Time limit:** 1 secor
**Memory limit:** 1024 ME

**Author:** Ryan Henning
**Source:** ICPC 2012 Nortl American Qualifier

Summer in Texas can be very hot. But Texans are tough, and in a weird way some of them enjoy the heat; for even the heat is bigger in Texas. But the heat can be quite uncomfortable for computer science students who are new to Texas and are not used to the sweating it can cause.

These students want to minimize the amount they sweat while walking from their dormitory to class. When they enter the sunshine, they begin sweating at the rate $r_0$ gallons/hour (where $r_0 > 0$). The longer they stay in the sun, the more they sweat. They sweat at a rate proportional to the square of how long they have been continuously exposed to the sun. Put another way, if they are exposed to the sun for twice as long, they sweat four times as much. But if they find a shady spot along the way, their continuous sun exposure is broken and they stop sweating immediately. They can then sit in the shade and cool off completely before continuing on their journey to class. Of course, leaving the shade means they begin sweating again, but starting at the lower rate $r_0$ (since they have cooled off).

Write a program that helps a student find a path from the dormitory to class which minimizes the total amount of sweat she expends.

## Input

Input describes the locations of shady spots on campus as well as the student's dormitory and class locations. Input begins with a line containing an integer $0 \leq n \leq 2500$, the number of shady spots. Each of the next $n$ lines contains a pair of integers $x$ $y$ specifying the coordinates of a shady spot. No two shady spots have the same coordinates. Following the shady spots are two more lines in the same format which specify the coordinates of the student's dormitory and class, respectively.

## Output

Print a path the student can take to get from her dormitory to class. The path should minimize the total sweat produced along the whole path. Print the path as indexes of the shady spots (from the order given in the input, with the first shady spot having index 0). If the best path contains no shady spots, output a single '-'. If there are multiple paths that minimize the total sweat, print any one of them.

### Sample Input 1

```
3
1 1
2 -2
5 -1
0 0
9 0
```

### Sample Output 1

```
1
2
```

### Sample Input 2

```
6
8 2
4 0
8 0
4 -1
7 -1
6 -2
2 1
9 2
```

### Sample Output 2

```
1
3
5
4
2
```

### Sample Input 3

```
1
-5 -5
0 0
10 10
```

### Sample Output 3

```
-
```