CCC 2012 Stage 2
Day 2, Problem 1: Editor Distance

Problem Description

You are using an editor to type in a program that you hope solves another problem on this contest. Instead of thinking about solving this problem, you look at your editor and start to think of how to edit your program more quickly.

Your editor has the following characteristics:

- you can move your cursor using the direction keys: up (↑), down (↓) , left (←) or right (→)
- pressing → will move the cursor one character to the right; if the cursor is on the rightmost character of a line, the cursor will move to the first character of the next line below the current line; (the cursor will not move if it is in the bottom-right position)
- pressing ← will move the cursor one character to the left; if the cursor is on the leftmost character of a line, the cursor will move to the last character of the previous line above the current line; (the cursor will not move if it is in the top-left position)
- pressing ↑ will move the cursor to the character immediately above it; if there is no character immediately above the cursor, it will move to the last character of the previous line above the current line; (the cursor will not move if it is on the first line)
- pressing ↓ will move the cursor to the character immediately below it; if there is no character immediately below the cursor, it will move to the last character of the next line below the current line; (the cursor will not move if it is on the last line)

You would like to find the least number of key presses that will cause you to move between positions in your editor in a given program which you are editing.

Input Specification

The first line of input is N, the number of lines of your program ($1 \leq N \leq 100000$). The next N lines contain the number of characters on each line: you can assume there is at least one character per line, and at most 80 characters per line. The next line contains two integers R_S C_S, indicating the starting row and column of the cursor ($1 \leq R_S \leq N$, $1 \leq C_S$, and C_S is at most the number of characters in row R_S). The last line contains two integers R_F C_F, indicating the finishing row and column of the cursor ($1 \leq R_F \leq N$, $1 \leq C_F$, and C_F is at most the number of characters in row R_F).

Output Specification

Output the minimum number of key presses that are required to move the cursor from row R_S and column C_S to row R_F and column C_F.

Sample Input

```
4
40
10
4
80
4 78
1 35
```

Output for Sample Input

```
10
```

CCC 2012 Stage 2
Day 2, Problem 2: Sample Size

Problem Description

Someone whose name I don't want to remember was reading an article in the newspaper that mentioned the results of a poll. Upon seeing that the percentages that appeared in the article were all 25%, 50% and 75%, they thought that maybe the poll only included 4 people, and was then not very good.

In this problem, your task is to come up with a program that would make a similar analysis. In particular, given the percentages that appear in a newspaper article, it is your task to determine the minimum number of people that must have been interviewed in order for those percentages to be possible, assuming that percentages are rounded to the closest integer, with .5 rounded up (so, 1 of 3 people corresponds to 33%, 2 of 3 people corresponds to 67%, and 155 of 1000 people corresponds to 16%).

Input Specification

The first line of the input file will contain an integer M between 1 and 100000 (inclusive). This represents the number of percentages that appear in the newspaper article you are analyzing.

M lines will follow, each of them containing an integer percentage P, $0 \leq P \leq 100$.

Output Specification

An integer $N \geq 1$, the minimum number of people that must be interviewed for all of the percentages in the input to correspond to a fraction of those people.

Sample Input 1

```
3
25
50
75
```

Sample Output 1

```
4
```

Sample Input 2

```
2
33
67
```

Sample Output 2

```
3
```

CCC 2012 Stage 2
Day 2, Problem 3: The Winds Of War

Problem Description

Colonel Trapp is trapped! For several days he has been fighting General Position on a plateau and his mobile command unit is now stuck at (0, 0), on the edge of a cliff. But the winds are changing! The Colonel has a secret weapon up his sleeve: the "epsilon net." Your job, as the Colonel's chief optimization officer, is to determine the maximum advantage that a net can yield.

The epsilon net is a device that looks like a parachute, which you can launch to cover any convex shape. (A shape is convex when, for every pair p, q of points it contains, it also contains the entire line segment $\overline{pq}$.) The net shape must include the launch point $(0, 0)$.

The General has P enemy units stationed at fixed positions and the Colonel has T friendly units. The *advantage* of a particular net shape equals the number of enemy units it covers, minus the number of friendly units it covers. The General is not a unit.

You can assume that

- no three points (Trapp's position $(0, 0)$, enemy units, and friendly units) lie on a line,
- every two points have distinct x-coordinates and y-coordinates,
- all co-ordinates (x, y) of the units have $y > 0$,
- all co-ordinates are integers with absolute value at most 1000000000, and
- the total number $P + T$ of units is between 1 and 100.

Input Specification

The first line contains P and then T, separated by spaces. Subsequently there are P lines of the form xy giving the enemy units' co-ordinates, and then T lines giving the friendly units' co-ordinates.

Output Specification

Output a single line with the maximum possible advantage.

Sample Input

```
5 3
-8 4
-7 11
4 10
10 5
8 2
-5 7
-4 3
5 6
```

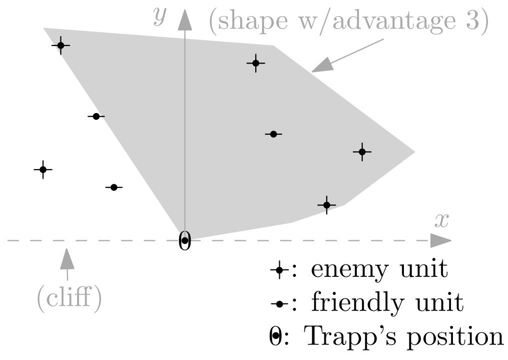

Figure 1: Sample input and an optimal net

Output for Sample Input

```
3
```