TimeDilation
Input File: TimeDilation.txt

Your older sister Nadia is part of the NASA deep space exploration team. As such she will be making fly-bys to stars and planets (e.g., Proxima Centauri, the closest star to our sun). The space ship she will use is a hyper-vehicle able to travel at speeds close to the speed of light. You would like to throw her a birthday party when she returns from her trip, but you are uncertain how old she will be when she returns because of a counterintuitive phenomenon called "time dilations".

Time dilations occur when traveling near the speed of light. The result of these dilations is to reduce the aging process of astronauts during their journey. Specifically, given the number of earth years a journey will take, N_J, we can calculate the number of years older an astronaut will be at the end of the journey, N_A, using the below formula:

$$N_A = N_J * \sqrt{(1 - v^2/c^2)}$$

where: v is the speed of the space ship, and c is the speed of light. So, if Nadia traveled on a 20 year journey at 0.995 times the speed of light the above equation reveals that she would only have only aged 2.0 years during the trip. If you were 19 and she was 25 when she began the journey, you would be 39 and she would be 27 when she returned. She would have become your *younger* sister!

Write a program to determine the number of years an astronaut will age on a given mission.

Inputs
The first line of input will contain the number of missions to process followed by the speed of light, your current age and Nadia's current age. This will be followed by one input line per mission that contains the ratio of the speed of the space ship to the speed of light, followed by the time (in earth years) for the round trip mission.

Outputs
For each mission output your age and Nadia's age when Nadia returns from the mission, followed by the speed of light and the speed of the space ship. Each mission's output should be on a separate line *formatted and annotated* (with commas and numeric precision) ***exactly*** as shown below.

Sample inputs
```
3 186282.27860 19.0 25.0
0.99500 20.0
0.10000 20.0
0.99990 20.0
```

Sample output
```
I will be 39.0 and Nadia will be 27.0, assuming: c = 186,282.28 v = 185,350.87
I will be 39.0 and Nadia will be 44.9, assuming: c = 186,282.28 v = 18,628.23
I will be 39.0 and Nadia will be 25.3, assuming: c = 186,282.28 v = 186,263.65
```

Tolls
Input File: Tolls.txt

Each square of a 10x10 checkerboard has a toll associated with it that must be paid when you enter the square. You wish to travel from the bottom most row to the top most row and

minimize the total of the tolls along the way. Write a program to output the row and column numbers of a route that minimizes the tolls. When making a move, you must stay on the board, the row number *must* increase by 1 and the column number can change by -1, 0, or +1. Tolls range from 0 to 9, and row 1 and column 1 is the lower left most square of the checkerboard.

	column 1	2	3	4	5	6	7	8	9	10
10	6	4	7	4	8	3	6	7	2	4
9	9	1	4	7	3	6	8	6	1	4
8	4	8	1	9	7	9	2	3	5	4
7	1	8	6	6	8	4	8	3	8	2
6	7	3	7	4	4	1	5	9	9	4
5	1	6	3	2	1	4	3	3	7	9
4	5	3	8	4	2	6	7	9	3	5
3	6	4	3	8	7	1	2	4	7	4
2	8	8	3	6	5	8	3	9	1	5
row 1	0	3	5	6	1	2	7	1	9	4

Inputs:

The tolls associated with each square of the checkerboard, one line per row. The first line of input is the tolls associated with row 10, the second line the tolls associated with row 9, etc. The tolls on a line will be separated by a space.

Outputs:

The minimum total toll followed by 10 lines that give the row and column numbers of the minimum toll path through the checkerboard. Each square's location will be on a separate line, with the row number preceding the column number. There will be a space between each row and column number.

Sample Input:

```
6 4 7 4 8 3 6 7 2 4
9 1 4 7 3 6 8 6 1 4
4 8 1 9 7 9 2 3 5 4
1 8 6 6 8 4 8 3 8 2
7 3 7 4 4 1 5 9 9 4
1 6 3 2 1 4 3 3 7 9
5 3 8 4 2 6 7 9 3 5
6 4 3 8 7 1 2 4 7 4
8 8 3 6 5 8 3 9 1 5
0 3 5 6 1 2 7 1 9 4
```

Sample Output:

```
23
1 8
2 7
3 6
```

```
4 5
5 5
6 6
7 7
8 8
9 9
10 9
```