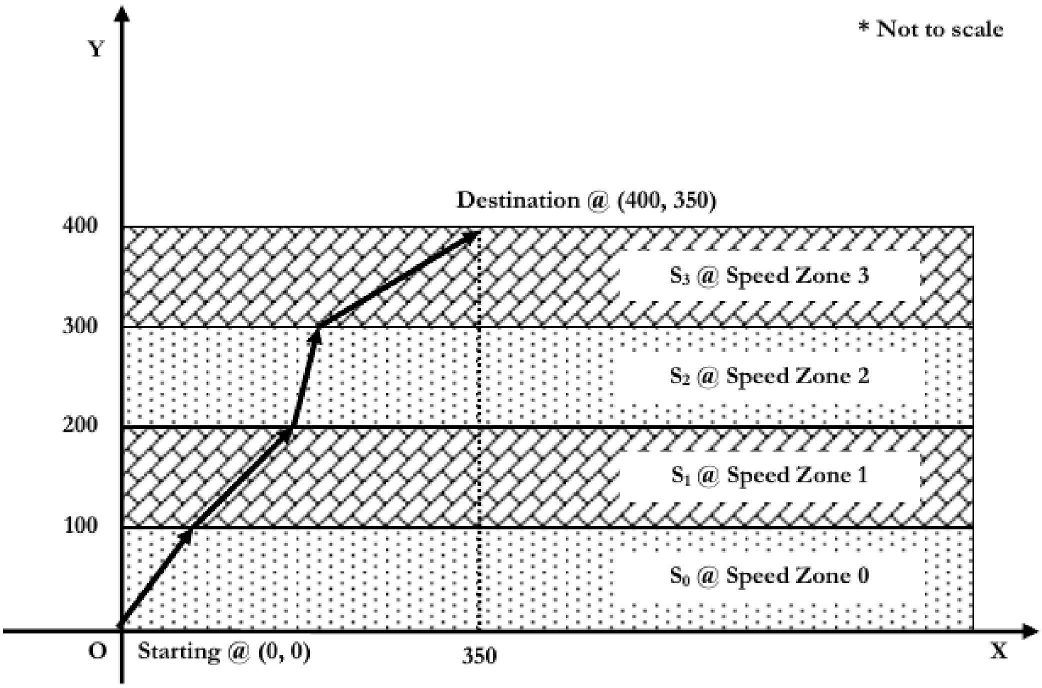
Suppose you are in a **2-Dimensional** world. Now, you are in a system of N parallel zones of **same** or **different speed**, numbered from 0 to $N-1$. In each zone you can move in some given constant speed (S_i amount per second in i-th zone) at any direction. Each zone is parallel to **X** axis, starting from the **X** axis (and then on the positive **X** and positive **Y** part only). Width of each zone is 100 (along the **Y** axis).

You are currently in the origin $(0,0)$. You need to reach $(100 * N, D)$ coordinate. But, you want to do that in minimum possible time (seconds).

Here is an example with $N = 4$, and $D = 350$. The arrows show **a possible path** from $(0,0)$ to $(400,350)$. Note that after the end of each zone (except the last one), it is possible that you may be in an **non-integer** '**X**' coordinate.



Given N, D, and the speeds S_0, S_1, S_2, …, S_{N-1} you will need to find the minimum possible time in seconds to reach the destination point.

Input

Input starts with an integer T (≤ 50), denoting the number of test cases.

Each case contains two lines. In the **first** line you will be given two integers N ($1 \leq N \leq 100$) and D ($0 \leq D \leq 10000$). In the **second** line you will be given N integers, the speeds, in the order: S_0, S_1, S_2, …, S_{N-1}. For all $0 \leq i < N$, $1 \leq S_i \leq 1000$.

Output

For each test case, generate one line of output, in the format '`Case` $< case - no >$`:` $< answer >$'. Here $< case - no >$ is the case number starting from 1, and $< answer >$ is the minimum possible time in seconds. Your output should not differ more than $10^{-6} = 0.000001$. You should print at least 8 digits after the decimal point for $< answer >$.

Sample Input

```
2
1 0
50
3 400
10 10 10
```

Sample Output

```
Case 1: 2.00000000
Case 2: 50.00000000
```