You are in a maze; seeing n doors in front of you in beginning. You can choose any door you like. The probability for choosing a door is equal for all doors.

If you choose the i-th door, it can either take you back to the same position where you begun in x_i minutes, or can take you out of the maze after x_i minutes. If you come back to the same position, you can remember last K doors you have chosen. And when you are about to choose a door, you never choose a door that is already visited by you. Or we can say that you never choose a door that is visited as one of the last K doors. And the probability of choosing any remaining door is equal.

Now you want to find the expected time to get out of the maze.

Input

Input starts with an integer T (≤ 100), denoting the number of test cases.

Each case contains a blank line and two integers n K ($1 \leq n \leq 100$, $0 \leq K \leq n$). The next line contains n space separated integers. If the i-th integer (x_i) is positive, you can assume that the i-th door will take you out of maze after x_i minutes. If it's negative, then the i-th door will take you back to the beginning position after $abs(x_i)$ minutes. You can safely assume that $1 \leq abs(x_i) \leq 10000$.

Output

For each case, print the case number and the expected time to get out of the maze. If it's impossible to get out of the maze, print '-1.000'. Otherwise print the result rounded to three places after the decimal point. Add 10^{-9} to your result to avoid precision errors.

Sample Input

```
4

2 0
10 10

2 0
10 -10

3 1
10 -10 -20

3 2
10 -10 -20
```

Sample Output

```
Case 1: 10.000
Case 2: 20.000
Case 3: 30.000
Case 4: 25.000
```