ETH Zurich
Institut für Theoretische Informatik
Prof. Dr. Angelika Steger
Prof. Dr. Emo Welzl
Prof. Dr. Peter Widmayer

Algorithms Lab

Exercise – *Magician and the Coin*

There is a new magician in town with a magic (of course, what else?) coin.

The magician stays in town for n days. On each day i, the magician flips his coin *once*, and you may choose how much money b_i you want to bet on the outcome of the coin flip. If the coin comes up heads you win and your wealth increases by b_i. If the coin comes up tails you lose and your wealth decreases by b_i.

The magic lies in the fact that on every given day i, the coin has probability exactly p_i of coming up heads, and $(1 - p_i)$ of coming up tails. A friend of yours knows the magician, and he tells you the probabilities p_i for all n days in advance. So you decide to test your luck. You start with a wealth of k, and your goal is to have a wealth of at least m in the end. Now, you wonder how you can maximise the probability of this outcome (to have at least a wealth of m in the end). With an optimal strategy, what is the probability of you having wealth at least m after n days?

Additional Information

- For your strategy, on the i-th day ($1 \leq i \leq n$) you should decide on some bet b_i for this day. Your choice may depend on the outcome of the coin flips on days $1, \ldots, i - 1$.
- The magician only accepts non-negative integer bets, so you are restricted to $b_i \in \mathbb{N}_0$. In particular, it is allowed to bet $b_i = 0$.
- You can never bet more money than you have. For example, on the first day you can choose any $b_1 \in \{0, \ldots, k\}$, but it is not possible to bet $k + 1$ or more.
- You only care whether or not you have wealth at least m in the end. It does not matter whether you end up with a wealth of 0 or of $m - 1$; both are failures.

Input The first line of the input contains the number $t \leq 30$ of test cases. Each of the t test cases is described as follows.

- It starts with a line that consists of three integers `n k m`, separated by a space. They denote
  - n, the number of days the magician stays in town ($1 \leq n \leq 10^2$);
  - k, your wealth on the first day ($0 \leq k \leq 10^2$);
  - m, the wealth you want to have by the time the magician leaves the town ($1 \leq m \leq 10^3$).
- The following line contains n real numbers `p`$_1$ `...` `p`$_n$, separated by a space, and such that $0 \leq p_i \leq 1$. Each p_i denotes the probability of winning on day i.

Output For each test case output a single line with the maximum probability of you having wealth at least m after n days. Each output value should be a real number between 0 and 1 rounded to *five* decimal places. You should round your result with the following piece of code:

```
std::cout << std::fixed << std::setprecision(5);
std::cout << 3.5 << std::endl; // Replace 3.5 with your desired double
```

Hint: We strongly advise you to use `double` throughout the algorithm for representing real numbers and not worry about the precision.

Points There are three groups of test sets, worth 100 points in total.

1. For the first group of test sets, worth 20 points, you may assume that $n \leq 5$ and $m \leq 10$. In addition, the probability of winning on any given day is the same and at most $1/2$, that is $p_1 = \ldots = p_n$ and $0 \leq p_i \leq 1/2$, for all $i \in [n]$.
2. For the second group of test sets, worth 30 points, you may assume that the probability of winning on any given day is the same and at most $1/2$, that is $p_1 = \ldots = p_n$ and $0 \leq p_i \leq 1/2$, for all $i \in [n]$.
3. For the third group of test sets, worth 30 points, there are no additional assumptions.
4. For the fourth group of test sets, which is hidden and worth 20 points, there are no additional assumptions.

Corresponding sample test sets are contained in `test`i`.in/out`, for $i \in \{1, 2, 3\}$.

Sample Input

```
3
2 2 8
0.5 0.5
4 2 33
0.25 0.5 0.75 1.0
5 2 20
0.3 0.5 0.2 0.7 1.0
```

Sample Output

```
0.25000
0.00000
0.12600
```

Explanation

Test Case 1: The only way to reach 8 is to bet all you have on each day.

Test Case 2: There is no way to reach 33 in only four days.

Test Case 3: An optimal strategy is to bet 2 on the first day, 1 on the second day (unless you are broke). If you win on the second day, you bet 0 on the third day, and bet all you have on the remaining two days. If your lose on the second day, you bet all you have on the remaining three days.