Back to Square 1

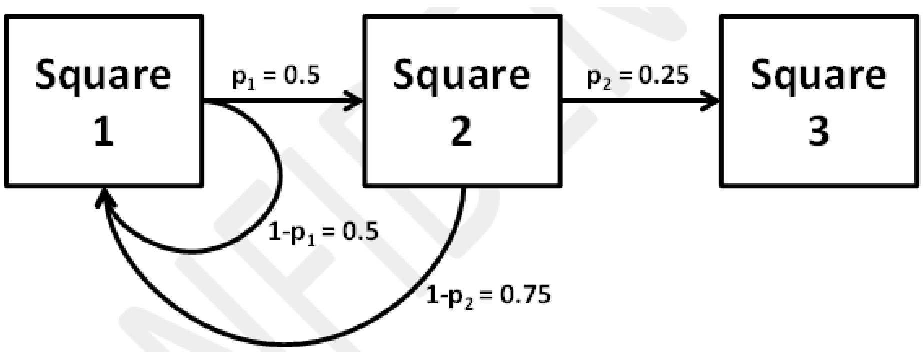
Problem Statement

Back to Square 1

The game “Back to Square 1” is played on a board that has *n* squares in a row and *n-1* probabilities. Players take turns playing. On their first turn, a player advances to square 1.After the first turn, if a player is on square *i* , the player advances to square *i* + 1 with probability *p(i)* , and returns to square 1 with probability 1-*p(i)* .The player is finished upon reaching square *n* .

Task

Write a program that determines the expected number of turns needed for a player to reach the final square.For example, consider the board below with *n* = 3 and *p(1)* = 0.5 and *p(2)* = 0.25. A player moves to square 1 on their first turn. With probability *p(1)* , they move to square 2 on their second turn, but with probability 1- *p(1)* , they remain on square 1. If they were lucky and made it to square 2 on their second turn, they advance to square 3 on their third turn with probability *p(2)* , but they would go back to square 1 with probability 1- *p(2)* . Thus, a really lucky player could finish is 3 turns. However, on average, it would take 13 turns for a player to make it to square 3.



Input

The input is made up of multiple test cases. Each test case contains 2 lines of input.
The first line in each test case is an integer *n* , 1 <= *n* <= 1,000, which represents the number of squares for this test case.
On the next line are *n* -1 single-space separated floating point numbers, each greater than 0 and less than or equal to 1, representing *p(1)* , *p(2)* , *p(3)* , ..., *p(n-1)* , respectively.
The input will end with a 0 on a line by itself.
Note: If for an input test case n=1 (i.e. there is only one square) then there will be no following line since there will be no probabilities. For example, the following input:
2
0.5
1
3
0.1 0.2
0
contains in total 3 test cases. The first one having 2 squares with an in-between transition

probability equal to 0.5, the second test case consists of a single square (and thus no transition probabilities are provided) and the last test case consists of 3 squares with respective transition probabilities equal to 0.1 and 0.2 .

Output

For each test case, output the expected number of turns needed to reach the final state, **rounded to the nearest integer**. You are guaranteed that the expected number of turns will be less than or equal to 1,000,000.
Note: Every line of output should end in a newline character .

Sample Input 1

```
3
0.5 0.25
0
```

Sample Output 1

```
13
```

Sample Input 2

```
2
0.5
4
0.3 0.2 0.1
0
```

Sample Output 2

```
3
228
```