Exercise 1

A Random Binary String

A random binary string contains n random bits. For $1 \le i \le n$, the probability that the i-th random bit is 1 is p_i, $0 \le p_i \le 1$. That is, the probability for the i-th random bit to be 0 is $1 - p_i$. Given n, another integer k, $2 \le k \le n \le 200$, and $p_1, p_2, p_3, \ldots, p_n$, your program should output the probability that the random binary string **does not** contain k (or more) consecutive 1's. For example, let n=4, p_1=0.9, p_2=0.8, p_3=0.7, p_4=0.6, and k=2, we have the following table.

Random binary string, *S*	Probability, *S* occurs	*S* Contains 2 (or more) consecutive 1's
0000	0.1×0.2×0.3×0.4=0.0024	
0001	0.1×0.2×0.3×0.6=0.0036	
0010	0.1×0.2×0.7×0.4=0.0056	
0011	0.1×0.2×0.7×0.6=0.0084	Yes
0100	0.1×0.8×0.3×0.4=0.0096	
0101	0.1×0.8×0.3×0.6=0.0144	
0110	0.1×0.8×0.7×0.4=0.0224	Yes
0111	0.1×0.8×0.7×0.6=0.0336	Yes
1000	0.9×0.2×0.3×0.4=0.0216	
1001	0.9×0.2×0.3×0.6=0.0324	
1010	0.9×0.2×0.7×0.4=0.0504	
1011	0.9×0.2×0.7×0.6=0.0756	Yes
1100	0.9×0.8×0.3×0.4=0.0864	Yes
1101	0.9×0.8×0.3×0.6=0.1296	Yes
1110	0.9×0.8×0.7×0.4=0.2016	Yes
1111	0.9×0.8×0.7×0.6=0.3024	Yes

The probability that the random binary string **does not** contain 2 or more consecutive 1's is 0.0024 + 0.0036 + 0.0056 + 0.0096 + 0.0144 + 0.0216 + 0.0324 + 0.0504 = 0.14. Note that since n and k can be as large as 200, your program must have an efficient algorithm in order to finish the computation in time.

Input Format

The input file contains several (at most 10) test cases. Each test case is given in a separated line with integers n, k, and probabilities $p_1, p_2, p_3, \ldots, p_n$. You can assume that $2 \le k \le n \le 200$ and no invalid probability value will be given as input. All integers and probabilities are separated by one or more space. The last test case will

be followed by a single line containing the integer 0.

Output Format

For each test case, your program should output a single line containing a floating point number to indicate the probability that the random binary string **does not** contain *k* (or more) consecutive 1's. The floating point number must be rounded to 5 digits after decimal point.

Sample Input

```
4 2 0.9 0.8 0.7 0.6
10 6 0 1 0.7 0.8 0.9 0.8 0.7 0.6 0.5 0.432
0
```

Sample Output

```
0.14000
0.66841
```