5863 Probability Computation

A random binary number X contains n independent random binary digits (bits) denoted by $b_1, b_2, b_3, \cdots, b_n$, where b_1 is the most significant bit and b_n is the least significant bit. That is, the value of X is $b_1 2^{n-1} + b_2 2^{n-2} + b_3 2^{n-3} + \cdots + b_n 2^0$. For each i, the random bit b_i is 1 with probability p_i percents ($0 \le p_i \le 100$) and b_i is 0 with probability $(100 - p_i)$ percents. Given the integer n ($1 \le n \le 200$), the integers $p_1, p_2, p_3, \cdots, p_n$, and the integers Q ($2 \le Q \le 99$) and R ($0 \le R < Q$), your program should output the probability of the event that X mod Q is equal to R, where mod is the modulus operation. In other words, your program should output the probability $Pr\{X \bmod Q = R\}$. The output probability must be rounded to 5 digits after the decimal point.

For example, consider a test case with $(n, p_1, p_2, p_3, p_4, Q, R) = (4,\ 0,\ 90,\ 100,\ 80,\ 5,\ 3)$. Your program should output 0.08000, since

$$
\begin{aligned}
Pr\{X \bmod 5 = 3\} &= Pr\{X = 3\} + Pr\{X = 8\} + Pr\{X = 13\} \\
&= Pr\{(b_1b_2b_3b_4) = (0011)\} + Pr\{(b_1b_2b_3b_4) = (1000)\} + Pr\{(b_1b_2b_3b_4) = (1101)\} \\
&= (100 - 0)\% \cdot (100 - 90)\% \cdot 100\% \cdot 80\% \\
&\quad + 0\% \cdot (100 - 90)\% \cdot (100 - 100)\% \cdot (100 - 80)\% \\
&\quad + 0\% \cdot 90\% \cdot (100 - 100)\% \cdot 80\% \\
&= 0.08000
\end{aligned}
$$

Note: The above example is for explanation. The straightforward algorithm in the example may not meet our time constraint when input integers are much larger. You should develop another **more efficient algorithm**.

Technical Specification

The ranges of input integers are: $1 \le n \le 200$, $0 \le p_i \le 100$ for each i, $2 \le Q \le 99$, and $0 \le R < Q$.

Input

The first line of the input file contains an integer indicating the number of test cases to follow. Then the input $(n, p_1, p_2, p_3, \cdots, p_n, Q, R)$ of each test case is given in a separated line. All integers are separated by one space.

Output

For each test case, your program should output the probability of the event that X mod Q is equal to R in a separate line. The probability must be rounded to 5 digits after the decimal point.

Sample Input

```
4
4 0 90 100 80 5 3
4 100 90 0 80 5 3
4 0 90 100 80 5 3
5 98 76 54 32 11 11 6
```

Sample Output

```
0.08000
0.74000
0.08000
0.25224
```