Problem A. Equipment Upgrade

Input file:	`standard input`
Output file:	`standard output`
Memory limit:	512 megabytes

Little Q is playing an RPG game. In this game, the weapon can be upgraded. Initially, the weapon is at level 0, and the upper bound of the level is n. Assume the current level is i ($0 \le i < n$), Little Q can pay c_i coins to upgrade the weapon, the next level will be $i+1$ with probability p_i, and will be $i-j$ ($1 \le j \le i$) with probability $(1-p_i)\frac{w_j}{\sum_{k=1}^{i} w_k}$.

Though Little Q is very rich, he is still wondering the expected number of coins for him to upgrade the weapon from level 0 to level n. Please write a program to help him.

Input

The first line contains a single integer T ($1 \le T \le 300$), the number of test cases. For each test case:

The first line contains a single integer n ($2 \le n \le 100\,000$), denoting the upper bound of the level.

The i-th ($1 \le i \le n$) of the following n lines contains two integers P_{i-1} and c_{i-1} ($1 \le P_{i-1}, c_{i-1} \le 100$), describing the success probability and the cost for level $i-1$. Here $p_{i-1} = \frac{P_{i-1}}{100}$. It is guaranteed that $P_0 = 100$.

The next line contains $n-1$ integers $w_1, w_2, \ldots, w_{n-1}$ ($1 \le w_i \le 100$).

It is guaranteed that the sum of all n is at most $500\,000$.

Output

For each test case, output a single line containing an integer, denoting the expected number of coins to level n.

More precisely, if the answer is $\frac{p}{q}$, you should output the minimum non-negative integer r such that $q \cdot r \equiv p \pmod{998\,244\,353}$. You may safely assume that such r always exists in all test cases.

Example

standard input	standard output
2 2 100 1 50 5 1 3 100 1 70 2 50 3 2 3	12 228170152