Problem F: Prize Lottery

You are at a game fair, where you play games to win tickets. To get value out of these tickets, a *Prize Lottery* has been set up. There are several prizes around the room that you can see (some golf clubs, a new car, a wristwatch, etc) , and each has a corresponding ticket bin.

At the end of the evening, after everyone has put all their tickets into bins, the prizes are drawn in sequence. For a given prize, a single ticket is drawn at random out of the corresponding ticket bin and the winner is revealed. The catch is that each person is only allowed to keep at most a single prize - if a person that has won a prize wins another, he may only keep one of the prizes (undoubtedly he'll take the one that's better...)

Prizes are indexed by i, and drawn in increasing order of i. Prize i has a value of V_i. You realize that the best strategy consists of waiting until everyone else has put their tickets into the bins and then putting yours in right at the end of the night. Consequently, you observe for each prize a number of other people's tickets N_i that are already in the ticket bin for this prize (you additionally note that everybody else has put their tickets into a single bin, and nobody else can possibly win more than 1 prize). You have K tickets, which of course you allocate optimally. What is the maximum expected value you can win?

Input Specification:

The input begins with an integer T, the number of test cases. Each test case consists of three lines. The first line contains positive integers $K \leq 125$, the number of tickets you possess, and $P \leq 125$, the total number of prizes. The next line consists of P non-negative integers representing V_i, none of which exceed 10000. The third line consists of P non-negative integers representing N_i, none of which exceed 1000.

Output Specification:

For each test case, output to five decimal places (rounded) the maximum expected value you can receive from the optimal ticket assignment.

Sample Input:

```
1
10 3
10 100 1
1 20 1
```

Sample Output:

```
34.48276
```