

Problem B. Those Russian Hackers

Input file: *standard input*
Output file: *standard output*
Time limit: 3 seconds
Memory limit: 256 mebibytes

During the day, you are just a usual software engineer, doing usual software engineering duties: you write code, drink coffee and attend meetings. But during the night, you become a legend that everybody is afraid of and, at the same time, everybody admires you. You are the famous Russian Hacker!

Several months ago, an important person asked you to rig an election in a small developing country far far away. Yesterday you logged into the main election server that is supposed to contain all the votes, and found out that during the election there will be a special security program which checks that nobody is connected to the server. To make the hackers' work even harder, it works in the following complicated manner.

The whole election lasts for $k \cdot n$ seconds. Time in this problem considered to be discrete: there are $k \cdot n$ units of time numbered from 0 to $k \cdot n - 1$, and any possible event lasts for an integer number of these units.

The time of election is divided into k equal time frames: seconds 0 to $n - 1$, seconds n to $2n - 1$, and so on. The security program starts working exactly with the beginning of election, and it is going to perform k security checks, one during each of the time frames. Each security check lasts exactly one second. For each check, the particular second when it happens is selected randomly using the probability distribution $p_0, p_1, \dots, p_{n-1}$. Formally, a security check will happen during the second $i \cdot n + j$ with probability p_j . The security check times for different time frames are independent from each other.

You may log into the system at any second, then you have to spend some time that is needed to perform your hacking business. You know the hacking time depends on lots of factors, and may be considered as an integer random value defined by the probability distribution $q_1, q_2, \dots, q_{2n-2}$. Formally, the hacking process will take j seconds with probability q_j . Of course, this random value is independent from the security check times. You have only one attempt of logging in, so once logged in, you have to finish your job.

Fortunately, you succeeded to install a small backdoor program that sends you a message each time a security check is finished, and you may use this information to plan your further attack. In particular, you may choose to log in immediately after the message, at the next second after a security check happens.

If during any second when you are logged into the election server, a security check happens, the election will be interrupted, meaning that you have failed. Also, if you don't finish your job during the election time, you also fail.

What is the probability of your success if you choose the best possible strategy?

Input

The first line contains two integers n and k ($2 \leq n \leq 300\,000$, $1 \leq k \leq 300\,000$): the length of the time frame that each security check belongs to and the number of security checks.

The second line contains n space-separated real values $p_0, p_1, \dots, p_{n-1}$ ($0 \leq p_i \leq 1$, $\sum_{i=0}^{n-1} p_i = 1$) that define the distribution of position of the exact time of security check during its time frame.

The third line contains $2n - 2$ space-separated real values $q_1, q_2, \dots, q_{2n-2}$ ($0 \leq q_i \leq 1$, $\sum_{i=1}^{2n-2} q_i = 1$) that define the distribution of the time you need to hack the election results.

All probabilities are given with at most six digits after the decimal point.

Output

Output the maximum probability of success that you may get by choosing an appropriate strategy. Your result will be considered correct if its absolute error does not exceed 10^{-6} .

Examples

standard input	standard output
3 1 0.5 0.25 0.25 0.75 0.25 0.0 0.0	0.6875000000
3 2 0.3 0.4 0.3 0.0 0.0 0.0 1.0	0.0900000000
3 1 0.5 0.2 0.3 1.0 0.0 0.0 0.0	0.8000000000

Note

In the first sample case, there will be only one security check, and the best possible strategy is to wait until it is finished, then log in and try to hack the election before the time runs out. The probability of success equals to $0.5 \cdot (0.75 + 0.25) + 0.25 \cdot 0.75 + 0.25 \cdot 0 = 0.6875$.

In the second sample case, you know that in any case you have to spend exactly four seconds hacking, and this means that you have to log in and start hacking at second 1 and finish by second 4 irrespective of the security checks. You succeed only if the first check happens at second 0 and the second check happens at second 5. The probability of success is therefore $0.3 \cdot 0.3 = 0.09$.

In the third sample case the best strategy is to always hack during second 1 that leads to a success probability of 0.8.