

G. Raffles

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Johnny is at a carnival which has n raffles. Raffle i has a prize with value p_i . Each participant can put tickets in whichever raffles they choose (they may have more than one ticket in a single raffle). At the end of the carnival, one ticket is selected at random from each raffle, and the owner of the ticket wins the associated prize. A single person can win multiple prizes from different raffles.

However, county rules prevent any one participant from owning more than half the tickets in a single raffle, i.e. putting more tickets in the raffle than all the other participants combined. To help combat this (and possibly win some prizes), the organizers started by placing a single ticket in each raffle, which they will never remove.

Johnny bought t tickets and is wondering where to place them. Currently, there are a total of l_i tickets in the i -th raffle. He watches as other participants place tickets and modify their decisions and, at every moment in time, wants to know how much he can possibly earn. Find the maximum possible expected value of Johnny's winnings at each moment if he distributes his tickets optimally. Johnny may redistribute all of his tickets arbitrarily between each update, but he may not place more than t tickets total or have more tickets in a single raffle than all other participants combined.

Input

The first line contains two integers n , t , and q ($1 \leq n, t, q \leq 200\,000$) — the number of raffles, the number of tickets Johnny has, and the total number of updates, respectively.

The second line contains n space-separated integers p_i ($1 \leq p_i \leq 1000$) — the value of the i -th prize.

The third line contains n space-separated integers l_i ($1 \leq l_i \leq 1000$) — the number of tickets initially in the i -th raffle.

The last q lines contain the descriptions of the updates. Each description contains two integers t_k, r_k ($1 \leq t_k \leq 2, 1 \leq r_k \leq n$) — the type of the update and the raffle number. An update of type 1 represents another participant adding a ticket to raffle r_k . An update of type 2 represents another participant removing a ticket from raffle r_k .

It is guaranteed that, after each update, each raffle has at least 1 ticket (not including Johnny's) in it.

Output

Print q lines, each containing a single real number — the maximum expected value of Johnny's winnings after the k -th update. Your answer will be considered correct if its absolute or relative error does not exceed 10^{-6} .

Namely: let's assume that your answer is a , and the answer of the jury is b . The checker program will consider your answer correct, if .

Examples

input
2 1 3 4 5 1 2 1 1 1 2 2 1
output
1.666666667 1.333333333 2.000000000

input
3 20 5 6 8 10 6 6 6 1 1 1 2 1 3 2 3 2 3
output
12.000000000 12.000000000 11.769230769 12.000000000 12.000000000

Note

In the first case, Johnny only has one ticket to distribute. The prizes are worth 4 and 5, and the raffles initially have 1 and 2 tickets, respectively. After the first update, each raffle has 2 tickets, so Johnny has expected value of winning by placing his ticket into the second raffle. The second update adds a ticket to the second raffle, so Johnny can win in the first raffle. After the final update, Johnny keeps his ticket in the first raffle and wins .

In the second case, Johnny has more tickets than he is allowed to spend. In particular, after the first update, there are 7, 6, and 6 tickets in each raffle, respectively, so Johnny can only put in 19 tickets, winning each prize with probability . Also, note that after the last two updates, Johnny must remove a ticket from the last raffle in order to stay under the tickets in the third raffle.