J: Caves

Memory limit **256 MB**

Sophie is a treasure seeker. Right now she's looking for a treasure hidden in a system of n caves, which are numbered from 1 to n. She knows that the cave system contains exactly one treasure in one of the caves. Sophie has done some measurements using a georadar; this has not told her where the treasure is, but she has obtained, for each cave, a probability that the treasure is in that cave.

Sophie will continue her search using a precise remote probe: after the probe enters a cave, it instantly finds out whether the cave contains the treasure. The probe starts its search in cave number 1, and it is known that this cave does not contain the treasure. Sophie knows (from her georadar measurements) which pairs of caves are connected using bidirectional tunnels which the probe can traverse. Such a traversal takes the probe one minute. Unfortunately, there is no guarantee that the probe can make its way from cave 1 to every other cave (even in many steps).

As an alternative to traversing the tunnels, Sophie can choose to abandon her probe and insert a new probe into the system, which takes t minutes. If she does so, communication with the old probe is lost and Sophie can no longer use it. Under these field conditions, inserting a new probe is very imprecise: it lands with uniform $\frac{1}{n}$ probability in any of the n caves. Sophie instantly knows where the probe has landed. Sophie has infinitely many probes at her disposal.

Help Sophie to find the treasure fast. Find a program which reads the size of the cave system, a description of the tunnels, the time t of introducing a new probe, as well as the probability distribution of where the treasure is, computes the smallest (over all Sophie's strategies) expected time of finding the treasure, and writes the computed value to standard output. By finding the treasure we mean reaching the treasure cell with a probe.

Input

The first line of input contains three space-separated integers n, m, t ($2 \le n \le 20$, $0 \le m \le \frac{n(n+1)}{2}$, $1 \le t \le 100$). They denote the number of caves, the number of tunnels and the time needed to insert a new probe. The second line contains n space-separated real numbers $p_1, p_2, \ldots, p_n$ ($p_1 = 0$, $0 \le p_i \le 1$, $\sum_{i=1}^{n} p_i = 1$); the number p_i is the probability that the treasure is in the i-th cave. These values are given with two decimal digits of precision. The next m lines describe the tunnels. The description of each tunnel consists of two space-separated integers i, j ($1 \le i, j \le n$ and $i \ne j$), which denote a tunnel between caves i and j that the probe can traverse in either direction. You can assume that all tunnels are distinct.

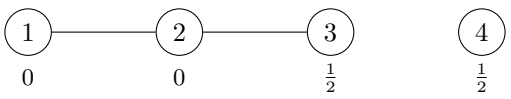
Output

The first and only line of output should contain a real number — the minimum possible expected time of finding the treasure.

The answer is considered correct if its absolute or relative error is at most 10^{-6}.

Example

Input	Output
4 2 10 0.00 0.00 0.50 0.50 1 2 2 3	22.000000000



Sophie should first move the probe to cave 2 and then to cave 3. With probability $\frac{1}{2}$ it has found the treasure in time 2. If it has not, she should now start inserting new probes, until one gets inserted into the cave 4. The expected number of insertions is 4, so the expected total time in this case is 42. On average, it takes Sophie 22 seconds to find the treasure. (Note that after 2 minutes, Sophie already knows where the treasure is, but she still needs to reach it with a probe.)

Input	Output
6 5 2 0.00 0.00 0.00 0.00 0.50 0.50 1 2 2 3 3 4 4 5 5 6	4.100000000