Toll Gates on a Tight Schedule

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	1024 megabytes

You have to cross the city before the last train leaves.

The city has n junctions and m one-way roads. Each road takes exactly one minute to pass. Junction i has a toll gate, and passing through it costs c_i coins.

You start at junction 1 and want to reach junction n. Since you are in a hurry, you will only consider routes that use the minimum possible number of roads.

Among all such shortest routes, count how many have total toll at most C. The total toll of a route is the sum of c_i over all junctions on the route, including junctions 1 and n.

Because the answer may be large, output it modulo 2^{64}.

Input

The first line contains three integers n, m, and C ($1 \le n \le 90$, $0 \le m \le 300$, $0 \le C \le 10^{12}$).

The second line contains n integers $c_1, c_2, \ldots, c_n$ ($0 \le c_i \le 10^{12}$), where c_i is the toll of junction i.

Each of the next m lines contains two integers u and v ($1 \le u, v \le n$, $u \ne v$), denoting a one-way road from u to v.

There are no repeated directed roads.

Output

Print one integer: the number of shortest routes from junction 1 to junction n whose total toll is at most C, modulo 2^{64}.

Example

standard input	standard output
4 5 8 1 2 3 4 1 2 1 3 2 4 3 4 2 3	2

Note

In the example, the shortest routes from junction 1 to junction 4 use exactly two roads.

They are $1 \to 2 \to 4$, with total toll $1 + 2 + 4 = 7$, and $1 \to 3 \to 4$, with total toll $1 + 3 + 4 = 8$.

Both routes have total toll at most 8.

The route $1 \to 2 \to 3 \to 4$ is not counted, because it is not a shortest route.