

Problem C

Costly Contest

The company Mindsight is holding a programming contest for n contestants of varying ages. It has been decided that the contest will be separated into k age divisions. The duration of the contest will be t minutes, the same for all divisions. Mindsight has created a pool of m available problems for use in the contest, and since all divisions will compete at the same time, the same problems can be used in multiple divisions without any issues.

In each division, the participant that solves the largest number of problems gets a prize, and in case of a tie (same number of problems solved) everyone tied for first place gets a prize. In particular, if no one in a division solves any problems, then everyone in that division gets a prize.

Mindsight has now come to the horrible realization that with these rules it is possible that all participants win a prize. This could bankrupt the company! So the company has enlisted a team of experts to help resolve the situation.

The expert group analyzed the data in depth. For each of the n participants, their skill level was quantified: for the i 'th participant, a *slowness factor* s_i was determined. Then, for each of the m problems available, its difficulty was quantified: for the j 'th problem, a *difficulty rating* d_j was assigned. The experts predict that the time it takes for participant i to solve problem j is $s_i \cdot d_j$ minutes. Furthermore participants cannot work on multiple problems in parallel so the time it takes to solve multiple problem is the sum of times of solving the individual problems. It is also well-established that participants always solve problems in increasing order of difficulty, starting with the easiest problem.

Now it is up to you, the underpaid intern, to configure the divisions so as to minimize the number of awarded prizes. Your task is to partition the n participants into k non-empty age divisions, and for each age division choose a non-empty subset of the m available problems to use in that division. Each division must correspond to a *contiguous* age range of participants (e.g. a division *cannot* be “20-25 or 30-35 years old”). Recall that the same problem may be used in multiple divisions.

Input

The first line of input contains four integers n, m, k, t ($1 \leq n \leq 10^5, 1 \leq m \leq 100, 1 \leq k \leq n, 1 \leq t \leq 10^5$) – the number of participants n , the number of available problems m , the number of age divisions k , and the duration of the contest t . The second line contains n integers s_1, \dots, s_n the slowness factors of the participants ($1 \leq s_i \leq 10^5$). The participants are ordered by age, and you can assume no two participants have the same exact age. The third line contains m integers d_1, \dots, d_m the difficulty ratings of the problems ($1 \leq d_j \leq 10^5$).

Output

Output a single integer, the minimum number of prize winners.

Sample Input 1

```
4 3 2 10
2 4 3 4
11 3 6
```

Sample Output 1

```
2
```

Sample Input 2

4	3	1	10
4	2	2	6
2	4	4	

Sample Output 2

2
