DMOPC '19 Contest 2 P4 - A Greedy Problem

Time limit: 0.1s **Memory limit:** 128M
Java: 0.6s
PyPy 2: 0.3s
PyPy 3: 0.3s

Jack is doing a programming contest. There are N problems in this contest and the contest will last a total of T minutes. The i-th problem will take Jack exactly t_i minutes to solve. Having an aversion to certain problem types such as dynamic programming, Jack wonders how many subsets of problems he can solve within q_i minutes if he decides that he definitely wants to solve problem a_i and problem b_i. More formally, Jack wants to determine the number of subsets of problems that contain problem a_i and problem b_i whose sum of solve times is less than or equal to q_i. Since this number may be very large, output it modulo $10^9 + 7$. Two subsets are different if there is a problem in one of the subsets that does not appear in the other subset.

Help Jack answer a total of Q of these queries.

Constraints

In all subtasks,
$2 \leq N, T, Q \leq 2\,000$
$1 \leq t_i, q_i \leq T$
$1 \leq a_i, b_i \leq N$
$a_i \neq b_i$

Subtask 1 [20%]

$N, T \leq 100$

Subtask 2 [20%]

$Q \leq 10$

Subtask 3 [60%]

No additional constraints.

Input Specification

The first line contains three space-separated integers, $N\ T\ Q$.
The next line contains N space-separated integers, $t_1, t_2, \ldots, t_N$.
The next Q lines contain three space-separated integers, $a_i\ b_i\ q_i$.

Output Specification

Output Q lines. The i-th line should contain the answer to the i-th query.

Sample Input

```
3 90 2
20 30 40
1 2 50
1 3 90
```

Sample Output

```
1
2
```