Problem L
Last Orders

You are attempting a legendary pub crawl across town.

The town has k junctions which are connected by m roads. There is a pub at each junction. Each road could be traveled in both directions and has a non-negative travel time. There could be multiple roads between the same pair of junctions. You start the pub crawl at time 0 in pub 1.

Pub i is located at junction i and will close at absolute time c_i. A pint must be finished at or before closing time in order to count.

Your drinking ability changes as the night goes on: The 1st pint takes t_1 time units, The 2nd takes t_2, and so on, with the r-th pint taking t_r. Please note that t_i are not necessarily increasing.

You may visit pubs in any order (not necessarily starting by *drinking* at pub 1) and may return to the same pub multiple times, but you cannot drink two consecutive pints in the same pub.

Your task is to maximize the number of pints consumed before closing times run out.

Input

- One line containing the number of pints you are capable of drinking, r ($1 \leq r \leq 100$).
- One line containing r integers: the number of seconds taken to drink the pints, $t_1 \ldots t_r$ ($1 \leq t_i \leq 3600$).
- One line containing the number of pubs, k ($1 \leq k \leq 300$).
- One line containing k integers, the integer closing times $c_1 \ldots c_k$ of the pubs in seconds ($1 \leq c \leq 86400$).
- One line containing the number of roads, m ($1 \leq m \leq 90000$).
- m further lines, each containing three integers: a, b, and d, indicating that there is a connection between pubs a and b taking d seconds to travel ($1 \leq (a \neq b) \leq k$; $1 \leq d \leq 86400$). Please note that there could be more than one road between same pair of pubs.

Output

Output an integer — the maximum number of pints you can drink before all of the pubs are closed.

Sample Input 1

```
8
60 120 180 240 300 360 420 480
2
900 1500
1
2 1 90
```

Sample Output 1

```
4
```

Sample Input 2

```
10
2 2 2 2 2 3 3 3 3 3
4
35 30 15 30
4
1 2 5
3 4 1
4 1 5
3 2 1
```

Sample Output 2

```
7
```