# D. Social Network

Polycarpus got an internship in one well-known social network. His test task is to count the number of unique users who have visited a social network during the day. Polycarpus was provided with information on all user requests for this time period. For each query, we know its time... and nothing else, because Polycarpus has already accidentally removed the user IDs corresponding to the requests from the database. Thus, it is now impossible to determine whether any two requests are made by the same person or by different people.

But wait, something is still known, because that day a record was achieved — $M$ simultaneous users online! In addition, Polycarpus believes that if a user made a request at second $s$, then he was online for $T$ seconds after that, that is, at seconds $s, s + 1, s + 2, ..., s + T - 1$. So, the user's time online can be calculated as the union of time intervals of the form $[s, s + T - 1]$ over all times $s$ of requests from him.

Guided by these thoughts, Polycarpus wants to assign a user ID to each request so that:

- the number of different users online did not exceed $M$ at any moment,
- at some second the number of distinct users online reached value $M$,
- the total number of users (the number of distinct identifiers) was as much as possible.

Help Polycarpus cope with the test.

## Input

The first line contains three integers $n$, $M$ and $T$ $(1 \le n, M \le 20\,000, 1 \le T \le 86400)$ — the number of queries, the record number of online users and the time when the user was online after a query was sent. Next $n$ lines contain the times of the queries in the format "hh:mm:ss", where hh are hours, mm are minutes, ss are seconds. The times of the queries follow in the non-decreasing order, some of them can coincide. It is guaranteed that all the times and even all the segments of type $[s, s + T - 1]$ are within one 24-hour range (from 00:00:00 to 23:59:59).

## Output

In the first line print number $R$ — the largest possible number of distinct users. The following $n$ lines should contain the user IDs for requests in the same order in which the requests are given in the input. User IDs must be integers from $1$ to $R$. The requests of the same user must correspond to the same identifiers, the requests of distinct users must correspond to distinct identifiers. If there are multiple solutions, print any of them. If there is no solution, print "No solution" (without the quotes).

## Examples

### input
```
4 2 10
17:05:53
17:05:58
17:06:01
22:39:47
```

### output
```
3
1
2
2
3
```

### input

```
1 2 86400
00:00:00
```

**output**

```
No solution
```

## Note

Consider the first sample. The user who sent the first request was online from 17:05:53 to 17:06:02, the user who sent the second request was online from 17:05:58 to 17:06:07, the user who sent the third request, was online from 17:06:01 to 17:06:10. Thus, these IDs cannot belong to three distinct users, because in that case all these users would be online, for example, at 17:06:01. That is impossible, because $M = 2$. That means that some two of these queries belonged to the same user. One of the correct variants is given in the answer to the sample. For it user $1$ was online from 17:05:53 to 17:06:02, user $2$ — from 17:05:58 to 17:06:10 (he sent the second and third queries), user $3$ — from 22:39:47 to 22:39:56.

In the second sample there is only one query. So, only one user visited the network within the 24-hour period and there couldn't be two users online on the network simultaneously. (The time the user spent online is the union of time intervals for requests, so users who didn't send requests could not be online in the network.)