

DMOPC '14 Contest 7 P6 - Trip

Toronto is a large city.

With so many bus routes crisscrossing every which way in no coherent fashion, trip planning can quickly become unmanageable. The TTC has a trip-planning app for phones, but it requires a data plan. You, being an excellent programmer with no data plan, have decided to remedy the situation for all the poor data-less souls like you — but mainly you.

You've decided you're going to write an offline trip planner!

Being an expert Google-r, you've found that the TTC provides data dumps of all their routes, so you don't have to ~~enslave~~ hire workers to map out the city for you. There are S bus stations, numbered $1 \dots S$, and R bus routes, numbered $1 \dots R$. A bus route is represented by a sequence of station ids, and is unidirectional. Each station will have a number of timetables posted — **one** for **each** route passing through it — listing the times of arrival of buses for a particular route. Of course, timetables are necessary since **buses do not circulate between stations at fixed time intervals**. An additional piece of common wisdom is that a route is a first-in-first-out system: the first bus to leave from the start of the route is the first to arrive at the end, with no buses leaving or entering the route midway.

Each timetable will contain a number of entries: the arrival times of buses in minute form (1440 minutes in a day). All stations on a route will have the same number of entries into their timetables for that particular route.

If a user of your brilliant app is to leave from station U at time T and wishes to arrive at station V , what is the shortest amount of time they have to spend traveling "The Better Way"? Unfortunately, some routes may be under repair and hence unavailable, so there may not always exist a route from U to V . A particularly painful journey may take **more** than one day.

Input Specification

- The first line of input will contain two space-separated integers S ($2 \leq S \leq 20$), R ($1 \leq R \leq 5$).
- The second line will contain three space-separated integers U, V ($1 \leq U, V \leq S$), T ($0 \leq T < 1440$).
- For the next R lines, line r will start with S_r ($2 \leq S_r \leq 100$), the number of stations on route r . The line will then contain S_r integers, the station ids on the route. **No bus will take longer than a day to traverse the entire route.**
- The next Q lines, where Q is the sum of the number of stations on each route ($Q = \sum_{i=1}^R S_i$) describe timetable data. Each line will begin with an integer r ($1 \leq r \leq R$) followed by an integer s ($1 \leq s \leq S$) and continuing with the integer T_{rn} ($1 \leq T_{rn} < 100$). The rest of the line will contain T_{rn} arrival times of buses on route r arriving at station s , in minute form. **All times in a timetable are unique.**

Output Specification

The minimum amount of time the trip will take in minutes, if all routes are chosen optimally. It's possible that it might be impossible to reach the destination using only bus routes. Since you [did not have a nice experience with the subway](#), if this is the case output `stay home`.

Sample Input 1

```
12 2
1 12 0
10 1 2 3 4 5 6 7 8 9 10
3 9 11 12
1 1 1 100
1 2 1 200
1 3 1 300
1 4 1 400
1 5 1 500
1 6 1 600
1 7 1 700
1 8 1 800
1 9 1 900
1 10 1 1000
2 9 1 0
2 11 1 100
2 12 1 0
```

Sample Output 1

```
2880
```

Explanation

The journey takes multiple days.

You must spend **900** minutes arriving at station **9**. From there, you must transfer to route **2**, since route **2** is the only route that has station **12** on its path. A bus arrives at station **9** for route **2** at time **0**, so we must wait $1440 - 900 = 540$ minutes until a bus arrives. The bus then arrives at station **11** at time **100**, and then takes another $1440 - 100$ minutes to arrive at station **12**.

Therefore, the total time the journey takes is $900 + 540 + 100 + 1340 = 2880$ minutes.

Sample Output 2

```
12 1
1 12 0
10 1 2 3 4 5 6 7 8 9 10
1 1 1 100
1 2 1 200
1 3 1 300
1 4 1 400
1 5 1 500
1 6 1 600
1 7 1 700
1 8 1 800
1 9 1 900
1 10 1 1000
```

Sample Output 2

```
stay home
```

Explanation

There is no way to get to station **12**, since station **12** is not on any given bus route.