Problem E: An Optimized Lunch

Filename: `lunch`
Time limit: `5 seconds`

After getting off the Hulk rollercoaster, you realize it is nearing the midpoint of your universal trip. Jacob, who is in your group, urges everyone to go on the Hollywood Rip Ride Rockit one last time before it shuts down permanently. The rest of the group seems to share the same opinion, but Justin points out that everyone should eat first before waiting in even more lines. Everyone seems to want to spend the most amount of time possible in the parks, so you want to minimize the combined amount of time you spend moving between the parks and getting lunch. Other groups catch wind of your plan, so help them figure out the best choice too!

Problem

Given the layout of Universal, as well as possible lunch locations, determine the shortest amount of time that a group can start at one specified ride, go to and eat lunch, followed by arriving at a second specified ride.

Input

Input will begin with a single integer **c** representing the number of test cases.

The first line of each test case contains 2 integers: **n** and **g**, representing the number of locations and the number of groups you must help, respectively.

There will be **n** lines that follow, the **i^{th}** of which contains a string **s_i** of only lowercase alphabetical characters, representing the name of the **i^{th}** location and an integer **t_i**, representing the estimated time it takes to eat at the **i^{th}** location, respectively.

Following that will be **n** lines, each of which contains **n** integers. The **j^{th}** integer in the **i^{th}** line is **m_{ij}**, which, if **m_{ij}** = -1, indicates there is no connection between locations **i** and **j**. Otherwise, it indicates there is a direct connection between locations **i** and **j** and it takes **m_{ij}** minutes to travel from location **i** to location **j**. It is guaranteed that **m_{ii},** or the value of the distance from a location to itself, will be 0. However, it is not guaranteed that **$m_{ij} = m_{ji}$.**

Finally, there will be **2g** lines that follow, with each pair of lines representing a group's query you must solve. The first line in each query will contain two strings consisting only of lowercase English letters ([a-z]), **a_i** and **b_i**, and an integer **d_i**. **a_i** and **b_i** represent the names of the starting and ending location of a group, respectively. **d_i** represents the number of places they are considering getting lunch at. The second line in each query will contain **d_i** strings representing the names of the locations the group is considering to get lunch at. It is guaranteed that each of these strings will be one of the previously listed strings in the list of ride locations.

Output

For each group, print one line with a single integer, representing the minimum amount of time it takes them to get from their starting location to their ending location while also getting lunch. It will always be possible to travel between the queried locations and grab lunch in between at one of the listed preferred locations.

Input Bounds and Corresponding Credit

100 Points
• $1 \le \mathbf{c} \le 35$ • $3 \le \mathbf{n} \le 300$ • $1 \le \mathbf{g} \le 10^3$ • $1 \le \lvert\mathbf{a}_i\rvert, \lvert\mathbf{b}_i\rvert, \lvert\mathbf{s}_i\rvert \le 15$ • $1 \le \mathbf{t}_i \le 10^6$ • $-1 \le \mathbf{m}_{ij} \le 10^6$ • $1 \le \mathbf{d} \le n - 2$

Samples

Input	Output
1 5 2 hulk 3 velocicoaster 4 threebroomstick 2 burgerking 4 ripriderockit 5 0 4 -1 -1 5 -1 0 3 2 -1 1 3 0 -1 4 4 -1 6 0 -1 -1 2 -1 3 0 hulk ripriderockit 2 threebroomstick burgerking velocicoaster ripriderockit 1 burgerking	13 15