

# Q1 - Mysterious Room

In order to help you get familiar with the dungeon homework, here is the practice for implementing rooms for the dungeon.

The player will start at room index 0, there are four types of rooms: rooms with treasure, rooms with ice monster, rooms with fire monster, and empty room. The player should handle each situation in each room.

There are two main tasks in this challenge. Firstly, you are given a bunch of room index followed by room type and whom their neighbors are. You have to link all the rooms together.

Second, you are given a bunch of paths to walk through. You have to print out the room index of the final position or print out -1 if an invalid path is provided. Besides, if the room isn't empty, print out the corresponding response. **If the player has a response in a room, this room will become empty.**

## Input Format

In the first line, you are given two integer  $m$   $n$  ( $1 \leq m \leq 10$ ,  $1 \leq n \leq 10$ ).  $m$  represents the total number of rooms, and  $n$  represents the number of steps for testing.

Firstly, for each room, there is a character followed by four integers, representing room type and UP, DOWN, LEFT, and RIGHT neighbors of this room. Character "t" means treasure room, "i" means ice monster, "f" means fire monster, and "e" means empty. Furthermore, each number shows the room index of the neighbor, and -1 represents there is no neighbor in this direction.

Secondly, for each path, there will be  $p$  characters follow up. Characters are either u, d, l, or r representing UP, DOWN, LEFT, and RIGHT walking direction.

## Constraints

We recommend you using linked list to link the room and virtual function to handle the response.

## Output Format

Each output line contains the index and the corresponding response of the current room. If there is a treasure in the room, print "ya". If there is an ice monster in the room, print "fire attack". If there is a fire monster in the room, print "ice attack". If the room is empty or the path is invalid, don't do anything(you should still print "-1" if the path is invalid). Note that **the index 0 room will always be empty.**

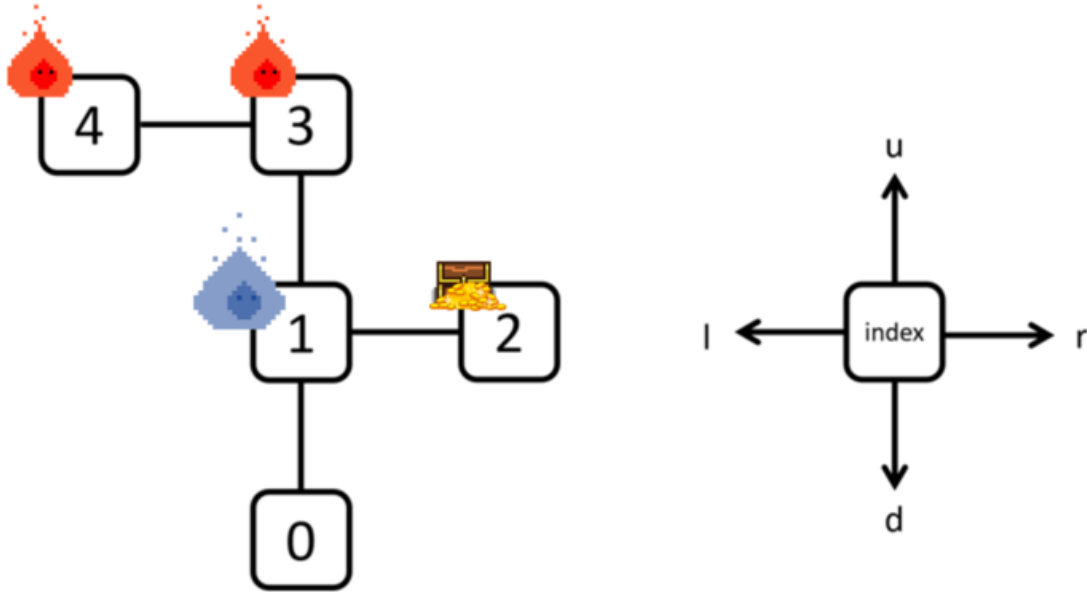
## Sample Input 0

```
5 6
e 1 -1 -1 -1
i 3 0 -1 2
t -1 -1 1 -1
f -1 1 4 -1
f -1 -1 -1 3
u r l l u l
```

## Sample Output 0

```
0
1 fire attack
2 ya
1
-1
3 ice attack
4 ice attack
```

## Explanation 0



You could link the room after create all the rooms or use `vector::reserve` if using vector.