

# Round 1

This section consists of 4 problems.

You have 1 hour and 45 minutes to work on these problems.

Your output must EXACTLY match the output given in the samples.

# The Lazy Time Traveler

*True laziness is achieved when one spends more effort avoiding work than the work would have taken in the first place.*

## Description

A very lazy time traveler lives on the equator of the Earth. Like many time travelers, he loves to travel! Paris, Cairo, Disney World — it would be a waste to only explore different times! Unfortunately, our Mr. Time Traveler suffers from crippling laziness. Lucky for him, his time machine can help! If he simply travels in time, and lets the Earth rotate underneath his time machine, he can force places to come to him — brilliant!

Your task is simple — given a starting location (in degrees), and a desired ending location (again, in degrees), calculate how long the time traveler will need to travel in time to arrive at his destination. Remember, the Earth spins  $360^\circ$  every 24 hours.

## Example Input/Output

You will be given a starting location  $s \in [0, 360]$  and an ending location  $e \in [0, 360]$ , separated by spaces. You must output a time, in hours between -12 and 12.

```
0 360
0.0

320 200
-8.0
```

# Chutes and SPACE-LADDERS

*While exploring the future, Seymore Time Traveler discovered a new technology — the LASER-AIDED-DIMENSIONAL-DRIFT-ENERGY-RIDER!*

## Description

Our hero is, yet again, in a predicament of laziness. He has discovered many exciting new worlds, connected together by *SPACE WORMHOLES*! He would love to travel through them...but he doesn't want to travel through *too* many wormholes (because that sounds hard).

Your task, should you choose to accept it, is to find the shortest path from the starting world to the destination of Seymore's choice.

## Example Input/Output

Each of the first  $n$  lines of input will be a world name (a single upper-case character), followed by a list of each connected world. . The  $n + 1$ 'th line will be a single upper-case letter, designating the desired destination. You must output a sequence of letters that represents the shortest path to the destination. Assume you start at the first given world. You can assume a path will always exist. There will be less than 10 worlds given. If multiple valid paths exist, any will be accepted.

```
input:
A B D
B C D
C
output:
A B C
```

```
input:
A B D
B C D
A
output:
A
```

```
input:
A B C D
B F
D B C
F E
E
output:
A B C E
```

# I'm Sorry I Killed an Ant and Created a Tyrant

*It's hard keeping track of what is and isn't your fault when traveling through time...or talking to women.*

## Description

As per the Code of Ethics of Time Travel, one must be careful with any changes that are made. When you're lazy, like Seymore, paying attention to all those silly effects (avalanche, domino, butterfly) is just too much effort so why not just do whatever and apologize when things go wrong later! What's the saying? It's easier to get forgiveness than attention?

Your task is potentially catastrophic — given a list of chain reactions resulting from something Seymore did (event A) and a worrying end case, determine if the end event was Mr. Traveler's fault so he can send back apology flowers.

## Example Input/Output

You will be given a variable number of lines consisting of upper-case letters. The first letter of each line is the causing event and each letter following it is a resulting effect. The last line in the list will have a single letter. You must determine if this event (letter) was caused by Mr. Traveler (event A). It will always be a directed acyclic graph.

```
A B
B C
C
true
A B
C D
D
false
```

# Querying Khan

*When Seymore Time Traveler has problems, he doesn't fret! He has friends all across the timeline to help him out!*

## Description

After visiting many of his friends across time, our favorite time traveler arrived at the dwelling of one Genghis Khan. After a night filled with tom-foolery and shenanigans, Seymore asked his friend for some help:

*Ok, Genghis, here is my problem. I have the DISTANCE between multiple points, but I don't actually know their location! Can you help me make a map of where the points must be?!*

To which Genghis responded by killing thousands.

## Example Input/Output

Each line of input will include a pair of labels, and their "Manhattan distance" between each other. You must print out a 10x10 table representing a valid grouping of these points, using \* to represent empty space, and a letter to represent the valid points.

```
input:
A B 5
A C 9
B C 5
output:
A*****
*****
*****
*****
B**C*****
*****
*****
*****
*****
*****
*****
*****
```

```
input:
A B 10
A C 6
A D 6
B C 5
B D 9
C D 5
output:
*****
*****
*****A**
*****
B**C*****
*****
*****D***
*****
*****
*****
*****
```