Practice Round APAC test 2016

Questions asked

**— Submissions**

Bad Horse

| 12pt | Not attempted **290/908 users** correct (32%) |
| 21pt | Not attempted **251/286 users** correct (88%) |

Captain Hammer

| 22pt | Not attempted **286/536 users** correct (53%) |

Moist

| 4pt | Not attempted **332/564 users** correct (59%) |
| 6pt | Not attempted **320/330 users** correct (97%) |

**— Top Scores**

| Piggietest | 65 |
| ivanzjj | 65 |
| wyh9346 | 65 |
| chx123456 | 65 |
| raypeng | 65 |
| fhector | 65 |
| mkrjn99 | 65 |
| thermal | 65 |
| milkbro | 65 |
| dtyfc | 65 |

## Problem A. **Bad Horse**

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the Quick-Start Guide to get started.

| Small input 1 12 points | Solve A-small-1 |
| Small input 2 21 points | Solve A-small-2 |

### Problem

As the leader of the Evil League of Evil, Bad Horse has a lot of problems to deal with. Most recently, there have been far too many arguments and far too much backstabbing in the League, so much so that Bad Horse has decided to split the league into two departments in order to separate troublesome members. Being the Thoroughbred of Sin, Bad Horse isn't about to spend his valuable time figuring out how to split the League members by himself. That what he's got you -- his loyal henchman -- for.

### Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each test case starts with a positive integer **M** on a line by itself -- the number of troublesome pairs of League members. The next **M** lines each contain a pair of names, separated by a single space.

### Output

For each test case, output one line containing "Case #x: y", where x is the case number (starting from 1) and y is either "Yes" or "No", depending on whether the League members mentioned in the input can be split into two groups with neither of the groups containing a troublesome pair.

### Limits

$1 \le T \le 100$.
Each member name will consist of only letters and the underscore character.
Names are case-sensitive.
No pair will appear more than once in the same test case.
Each pair will contain two distinct League members.

### Small dataset

$1 \le M \le 10$.

### Large dataset

$1 \le M \le 100$.

### Sample

| Input | Output |
|---|---|
| 2<br>1<br>Dead_Bowie Fake_Thomas_Jefferson<br>3<br>Dead_Bowie Fake_Thomas_Jefferson<br>Fake_Thomas_Jefferson Fury_Leika<br>Fury_Leika Dead_Bowie | Case #1: Yes<br>Case #2: No |

# apac test
powered by Google Code Jam

Questions asked

## Problem B. Captain Hammer

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the Quick-Start Guide to get started.

| Small input 22 points | Solve B-small |

### Problem

The Hamjet is a true marvel of aircraft engineering. It is a jet airplane with a single engine so powerful that it burns all of its fuel instantly during takeoff. The Hamjet doesn't have any wings because who needs them when the fuselage is made of a special Wonderflonium isotope that makes it impervious to harm.

Piloting the Hamjet is a not a job for your typical, meek-bodied superhero. That's why the Hamjet belongs to Captain Hammer, who is himself impervious to harm. The G-forces that the pilot endures when taking a trip in the Hamjet are legen-dary.

The Hamjet takes off at an angle of θ degrees up and a speed of **V** meters per second. **V** is a fixed value that is determined by the awesome power of the Hamjet engine and the capacity of its fuel tank. The destination is **D** meters away. Your job is to program the Hamjet's computer to calculate θ given **V** and **D**.

Fortunately, the Hamjet's Wondeflonium hull is impervious to air friction. Even more fortunately, the Hamjet doesn't fly too far or too high, so you can assume that the Earth is flat, and that the acceleration due to gravity is a constant 9.8 m/s$^2$ down.

### Input

The first line of the input gives the number of test cases, **T**. **T** lines follow. Each line will contain two positive integers -- **V** and **D**.

### Output

For each test case, output one line containing "Case #x: θ", where x is the case number (starting from 1) and θ is in degrees up from the the horizontal. If there are several possible answers, output the smallest positive one.

An answer will be considered correct if it is within 10$^{-6}$ of the exact answer, in absolute or relative error. See the FAQ for an explanation of what that means, and what formats of floating-point numbers we accept.

### Limits

1 ≤ **T** ≤ 4500;
1 ≤ **V** ≤ 300;
1 ≤ **D** ≤ 10000;
It is guaranteed that each test case will be solvable.

### Sample

```
Input          Output

3              Case #1: 45.0000000
98 980         Case #2: 15.0000000
98 490         Case #3: 3.8870928
299 1234
```

# Problem C. **Moist**

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the Quick-Start Guide to get started.

| Small input 1<br>4 points | Solve C-small-1 |
| Small input 2<br>6 points | Solve C-small-2 |

## Problem

Moist has a hobby -- collecting figure skating trading cards. His card collection has been growing, and it is now too large to keep in one disorganized pile. Moist needs to sort the cards in alphabetical order, so that he can find the cards that he wants on short notice whenever it is necessary.

The problem is -- Moist can't actually pick up the cards because they keep sliding out his hands, and the sweat causes permanent damage. Some of the cards are rather expensive, mind you. To facilitate the sorting, Moist has convinced Dr. Horrible to build him a sorting robot. However, in his rather horrible style, Dr. Horrible has decided to make the sorting robot charge Moist a fee of $1 whenever it has to move a trading card during the sorting process.

Moist has figured out that the robot's sorting mechanism is very primitive. It scans the deck of cards from top to bottom. Whenever it finds a card that is lexicographically smaller than the previous card, it moves that card to its correct place in the stack above. This operation costs $1, and the robot resumes scanning down towards the bottom of the deck, moving cards one by one until the entire deck is sorted in lexicographical order from top to bottom.

As wet luck would have it, Moist is almost broke, but keeping his trading cards in order is the only remaining joy in his miserable life. He needs to know how much it would cost him to use the robot to sort his deck of cards.

## Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each one starts with a line containing a single integer, **N**. The next **N** lines each contain the name of a figure skater, in order from the top of the deck to the bottom.

## Output

For each test case, output one line containing "Case #x: y", where x is the case number (starting from 1) and y is the number of dollars it would cost Moist to use the robot to sort his deck of trading cards.

## Limits

1 ≤ **T** ≤ 100.
Each name will consist of only letters and the space character.
Each name will contain at most 100 characters.
No name with start or end with a space.
No name will appear more than once in the same test case.
Lexicographically, the space character comes first, then come the upper case letters, then the lower case letters.

## Small dataset

1 ≤ **N** ≤ 10.

## Large dataset

1 ≤ **N** ≤ 100.

## Sample

```
Input                    Output

2                        Case #1: 1
2                        Case #2: 0
Oksana Baiul
Michelle Kwan
3
Elvis Stojko
Evgeni Plushenko
Kristi Yamaguchi
```