

UIL STATE 2020 COMPUTER SCIENCE

I. General Notes

1. Do the problems in any order you like. They do not have to be done in order from 1 to 12.
2. All problems have a value of 60 points.
3. There is no extraneous input. All input is exactly as specified in the problem. Unless specified by the problem, integer inputs will not have leading zeros. Unless otherwise specified, your program should read to the end of file.
4. Your program should not print extraneous output. Follow the form exactly as given in the problem.
5. A penalty of 5 points will be assessed each time that an incorrect solution is submitted. This penalty will only be assessed if a solution is ultimately judged as correct.

II. Point Values and Names of Problems

Number	Name
Problem 1	Dungeons and Dragons
Problem 2	From the Stars
Problem 3	Blocks
Problem 4	Steeds
Problem 5	Gems
Problem 6	Elven
Problem 7	Forest
Problem 8	Grey
Problem 9	Boxes
Problem 10	Booking
Problem 11	Doors
Problem 12	Currency

1. Dungeons and Dragons

Program Name: Dungeons.java**Input File: dungeons.dat**

After watching the lord of rings movies, you and your mates started playing dungeons and dragons, but you've noticed that your in-game backpack is running out of room. Write a program that will take stock of all the items you could carry, determine their value based on a given formula, and maximize the value you can carry, given the maximum weight that you can put into your backpack. The value formula is as follows:

For weapons: $V = (\text{damage per hit}) * (\text{your number of attacks per round} * 2)$

For Armor: $V = (\text{armor rating}) * (22 - (\text{your armor class}))$

For Items: $V = (\text{utility rating})$

Input

The first line will contain a single integer n that indicates the number of data sets that follow.

Each data set will begin with 4 integers, i , w , n , and a , denoting the number of items, the maximum weight you can carry in your backpack, the number of attacks you get per round, and your armor class. Each of the following i lines will contain a listing for an object. It will begin with two strings, the name of the object and its type (weapon armor item), followed by an integer, either the damage per hit, armor rating, or utility rating, depending on the object type, and then another integer, the object's weight.

Output

Output the string "Maximum value: ", followed by the maximum value you can get into your backpack.

Example Input File

```
2
5 12 2 19
Dagger weapon 8 4
Bow weapon 9 5
Chestplate armor 12 12
Helmet armor 9 10
Potion item 30 2
4 28 1 15
Sword weapon 12 8
Chainmail armor 10 10
Hammer weapon 10 9
Shield item 30 10
```

Example Output to Screen

```
98
124
```

2. From the Stars

Program Name: Star.java**Input File: star.dat**

You have been watching a lot of Lord of Rings lately and you're wondering about the hyperspace travel aspect of the universe. How exactly did Dumbledore land on earth and become Gandalf? How did Luke Skywalker travel from planet to planet? Write a program to calculate the shortest distances to travel between planets through hyperspace. Hyperspace is a bit finicky, so certain planets do not have hyperspace lanes between them, but most can. You will be given the 3D positions of all the planets, and a list of which planets are connected via hyperspace lanes.

Input

The input will begin with three integers, *p*, *n*, and *i*, denoting the number of planets, the number of entries in the list of pairs of planets with hyperspace lanes between them, and the number of pairs of planets to find the distance between. Each of the following *p* lines will contain a planet listing consisting of a string, the name of the planet, and three floating point numbers denoting the *x*, *y*, and *z* coordinates of the planet in the universe. The next *n* lines will each contain a pair of strings, the names of two planets, where there is a hyperspace lane. The following *i* lines will each contain a pair of planet names for you to determine the distance between.

Output

For each pair of planets that you were to find the distance between, output the two planet names in the order given, separated by the string " -> ", followed by a colon, another space, and the shortest possible distance between the two planets, rounded to 3 decimal places, with a comma separator when appropriate. If there is no path between the two planets, output the string "These are not the planets you're looking for."

Example Input File

```
6 6 2
Kamino 234.5 -109.4 34.7
Tatooine 927.43 85.34 -23.45
Hoth -34.56 -45.67 43.21
Geonosis -345.32 753.60 -456.03
Jakku 78.43 333.33 444.55
Endor 667.76 0.45 -1234.56
Kamino Endor
Jakku Endor
Hoth Tatooine
Geonosis Tatooine
Kamino Hoth
Tatooine Endor
Kamino Tatooine
Hoth Jakku
```

Example Output to Screen

```
Kamino -> Tatooine : 1,249.791
Hoth -> Jakku : 3,432.691
```

3. Blocks

Program Name: Blocks.java**Input File: blocks.dat**

Gargamel the Orc is obsessed with blocks. He is an idiot so may need your help. Mainly how numbered blocks can be stacked in order and rotated and flipped. He always starts with a 3x3 stack of 9 blocks, starting at 1 on the top left, and increasing in row major order to 9 in the bottom right like this:

```
1 2 3
4 5 6
7 8 9
```

He wants you to figure out a way to rotate the stack with just a number input system, each number corresponding to the amount of 90 degree turns, so an entrance of 1 would flip it 90 degrees onto its right side, and 2 would flip it 180 degrees upside down, and so on.

Input

The first line will contain a single integer n that indicates the number of data sets that follow. Each data set will include a single integer d that indicates the amount of 90 degree turns the above matrix will undergo.

Output

Output the matrix after it has d number of 90 degree turns along with 3 underscores as a divider.

Example Input File

```
4
0
1
2
3
```

Example Output to Screen

```
1 2 3
4 5 6
7 8 9
_ _ _
7 4 1
8 5 2
9 6 3
_ _ _
9 8 7
6 5 4
3 2 1
_ _ _
3 6 9
2 5 8
1 4 7
_ _ _
```

4. Steeds

Program Name: steeds.java**Input File: steeds.dat**

Gimli decided that he has had enough of the adventuring lifestyle, and he would much rather just settle down and start a farm. He decided that he will only raise horses though, and any other animal he will decide not to buy. Given a list of potential animals, output the number of animals Gimli buys, and the amount of money it will cost him.

Input

The first line will denote a single integer t that denotes the number of test cases that follow. For each test case, the first line will contain a single integer n that indicates the number of animals that follow. Each subsequent line will contain a single integer x denoting the price of the animal and a string s that denotes the type of animal.

Output

Output the amount of horses and the total price to buy all of them.

Example Input File

```
3
4
10 dog
11 horse
13 horse
4 cat
4
0 alien
14 horse
15 horse
6 horse
5
6 dog
6 cat
9 aardvark
6 horse
7 rat
```

Example Output to Screen

```
2 24
3 35
1 6
```

5. Gems

Program Name: gems.java**Input File: gems.dat**

Gimli's big brother, Gemli, is one of the most prolific miners in all of Middle Earth. He has an entire mining company based on the extraction of Morkite, a very unique gem located only in the dwarven mountain ranges. It has a very specific structure, and although individual gems range in size, the proportion of the gem always follows a specific pattern. Given a number describing the size of the gem, output the formation of the gem as shown in the output.

Input

The first line will contain a single integer n that indicates the number of data sets that follow. Each data set will start with a single integer x denoting the size of the gem to be outputted.

Output

Output the gem in the same format as shown below. Make sure to include an empty space between each test case.

Example Input File

```
3
1
3
5
```

Example Output to Screen

```
$

$
$$$
$

$
$$$
$$$$$
$$$
$
```

6. Elven

Program Name: elven.java**Input File: elven.dat**

Gimli's best friend Legolas is one of the best archers in all of Middle Earth. Gimli is trying to learn Elven, but he is having some trouble writing it when translating from standard Common (English). Written Elven is shockingly similar to Common, with the exception that the vowels (AEIOU) are replaced with lowercase versions of themselves, and the string is reversed. Given a sentence in common, output the Elven equivalent of the sentence.

Input

The first line will contain a single integer n that indicates the number of data sets that follow. Each data set will start with a single integer x denoting how many sentences are in each set. The following x lines will contain sentences in Common.

Output

Output the Elven equivalent to the Common sentence given. Make sure to include an empty line between each data set.

Example Input File

```
1
2
```

```
THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG.
JIMMY BUTLER WILL WIN THE NBA FINALS
```

Example Output to Screen

```
.GoD YZaL eHT ReVo SPMuJ XoF NWoRB KCiuQ eHT
SLaNiF aBN eHT NiW LLiW ReLTuB YMMiJ
```

7. Forest

Program Name: forest.java**Input File:** forest.dat

Legolas is returning home to the elven forests of Middle Earth, but he is quite bored and is trying to find a way to spend the time. On the back of his carriage, he decides to play a little scenery game: count up the amount of boulders, trees, and bushes on either side of the road and see which side has more of each object.

Input

The first line will contain a single integer n that indicates the number of test cases that follow. Each data set will start with 2 lines. The first line represents the left side of the road, and the second line represents the right side of the road. An "i" represents a tree, a "o" represents a boulder, a "m" represents a bush, and a "." represents empty space.

Output

Output which side of the road has more boulders, trees, and bushes, in the format shown below. There will never be a tie. Include an empty space between test cases.

Example Input File

```
3
oii.mom.
mmm..i..
ii..oi.mmm.i.o
iiii.ooo.mm...i.
iiiiiiiiiiiiiiiiiii
ooooooooooooooooom
```

Example Output to Screen

```
BOULDERS: LEFT
TREES: LEFT
BUSHES: RIGHT
```

```
BOULDERS: RIGHT
TREES: RIGHT
BUSHES: LEFT
```

```
BOULDERS: RIGHT
TREES: LEFT
BUSHES: RIGHT
```


8. Grey

Program Name: grey.java**Input File: grey.dat**

On Gimli's adventures, he decided to stop by Gandalf's tower in Middle Earth. After learning some Elven from Legolas, he decided to go learn some basic magic from Gandalf. To get Gimli started, Gandalf had Gimli learn from some of his beginner spellbooks. Gimli decides to learn a time travel spell. Certain crystals can enhance the power of the spell. Without crystals, the spell would send someone back in time a week. Crystals increase the power of the spell by an extra week. Given the date of the spell cast and the amount of crystals, calculate the day that Gimli would be sent to if he cast the spell correctly.

Input

The first line will contain a single integer n that indicates the number of test cases that follow. Each data set will contain a single date given in the format "MM/DD/YYYY", and a single integer k that represents the amount of crystals Gimli has for the spell.

Output

Output the theoretical date Gimli would be sent to if he casted the time spell.

Example Input File

```
3
01/22/2001 1
04/16/2005 2
11/11/2011 4
```

Example Output to Screen

```
01/15/2001
04/02/2005
10/14/2011
```

9. Boxes

Program Name: boxes.java**Input File: boxes.dat**

Gimli is now tasked to move some boxes around the city of Belegost. He will be paid handsomely for this, so he wants to make sure he does this right. He has big boxes, and he has small boxes. His employer wants him to move all of the big boxes to the western side of town, and he wants all of the small boxes to be moved to the eastern side of town.

Input

The first line will contain a single integer n that indicates the number of data sets that follow. Each data set will contain a single line that represents the landscape of Belegost. Big boxes are denoted with the letter "B" and small boxes are denoted with the letter "b". There may be other random characters inserted throughout the line.

Output

Output the line given, but with all the big boxes at the leftmost side of the line, all the small boxes to the rightmost side of the line, and the random characters inserted in the middle in the order it was inputted. There will always be at least one small box and one big box.

Example Input File

3

b..asiduhweiufhBBBBWQWiwueghb

bbBB

BasuhdiouashiduhwibbbBBBBBBBBBBbbarjhoqwijf

Example Output to Screen

BBBB..asiduhweiufhWQWiwueghbb

BBbb

BBBBBBBBBBBasuhdiouashiduhwiarjhoqwijfbbbbbb

10. Booking

Program Name: booking.java

Input File: booking.dat

Gimli decides to stay overnight at the local tavern in Belegost. The folks who run the place are looking for employees. Gimli offers and is given a job. His role is to sort the patrons of the tavern given a certain criteria and write their names down in that order into the ledger. Each patron entry contains their name and their room number. Gimli is to sort the patrons based on their room number in ascending order, then alphabetically (last name, then first) if there are multiple patrons with the same floor number.

Input

The first line will contain a single integer n that indicates the number of data sets that follow. Each data set will start with a single integer x denoting how many patrons are in each set. For each patron, there will be two strings denoting their first and last name separated by a space, and an integer k denoting the floor number of their room.

Output

Output the first and last name of the sorted patrons using the criteria above. Separate each data set with an empty line in the output.

Example Input File

```
3
4
Hudson Craven 1
Gustavo Alba 4
Irone Sola 3
Hudson River 1
2
Owen Giles 2
Owea Giles 2
1
Albert Venter 9
```

Example Output to Screen

```
Hudson Craven
Hudson River
Irone Sola
Gustavo Alba

Owea Giles
Owen Giles

Albert Venter
```

11. Doors

Program Name: Doors.java

Input File: doors.dat

You know game shows, where they have 3 doors and there's prizes behind each one? Well, this is nothing like that. You have been trapped in a maze and you need to escape ASAP so you can pick up your mom from the airport. The maze will be full of doors and keys to open those doors. You need to find out if it is possible to escape from the maze, or if your mom will be very angry with you, and if it is possible, how many steps you need to get out of the maze, if you move optimally. You will be given a map of the maze made up of the following characters:

- ' . ' – represents an open space in the maze that you can pass through.
- ' # ' – represents a wall, which is impassable.
- [' a ' - ' z '] – represents a key, with each letter corresponding to the door which is its uppercase counterpart.
- [' A ' - ' Z '] – represents a door, with each letter corresponding to the key which is its lowercase counterpart.
- '] ' – denotes your starting point within the maze.
- ' [' – denotes the exit from the maze, this can be at any position within the maze.

There will only be one copy of each keys, but there may be more than one of the same type of door. In other words, some keys may unlock multiple doors each. You may only move in the 4 cardinal directions (up, down, left, right). Each step takes 1 second, and if you have 36 seconds and it takes 36 seconds to escape, you make it. When passing over a spot where you've already picked up the key, treat it as an open space.

Input

The first line will contain a single integer n that indicates the number of data sets that follow.

Each data set will begin with 2 integers, r and c , denoting the number of rows and columns in the map of the maze, and the amount of time (in seconds) you have to escape the maze. The following r lines will each contain c characters of the map of the maze.

Output

If it is possible to escape the maze output "Amazeing Job. Escaped in [s] steps.", where [s] is replaced by the number of steps in the shortest path to escape the maze. If you cannot escape the maze output "Mother knows best.".

Example Input File

```
2
6 6 10
]..A.[
#...#.
..#..B
#..a#.
b#....
....##
5 5 8
[..#a
ABC..
b..#.
.#c.]
...##
```

UIL STATE 2020

Example Output to Screen

Mother knows best.

Amazeing Job. Escaped in 7 steps.

12. Currency

Program Name: currency.java**Input File: currency.dat**

Gimli realized one day that all the separate civilizations on Middle Earth all use their own currencies when purchasing goods, and he really didn't like that. He decided to use his own wits to convert a variety of other currencies. However, he has a problem. Some currencies don't have a direct conversion rate listed in his resources. Help him use other shared conversion rates to calculate the relationship between two given currencies. Note: By the nature of economics, if there are multiple different ways to convert one coin to another, they will inherently all be equal.

Input

The first line will contain a single integer n that indicates the number of data sets that follow. Each data set will start with a single integer x , the number of currencies that is contained in the data set. The next x lines will contain the names of different currencies. There will then be a single integer p , which denotes the number of conversions between two currencies. The next x lines will contain two strings, m and o , which are two previously stated currencies, and one decimal value, i , which represents the conversion rate between m and o (how many m 's will equal one o). There will be a final pair of strings, k and s , which is the desired conversion rate Gimli is trying to find. Note: Traversing from o to m is just multiplying o by the $1/i$.

Output

Output the desired conversion rate (Round to two decimal places). If there is no way to convert the two currencies, output "NO CONVERSION RATE AVAILABLE".

Example Input File

```
2
3
USD
JPY
GBP
4
USD JPY 110
JPY GBP 0.0070
USD GBP
3
GOLD
SILVER
COPPER
2
SILVER COPPER 10
SILVER GOLD .1
GOLD COPPER
```

Example Output to Screen

```
0.77
100.00
```