Dulles Computer Science Packet 2015-2016

Shreyas B. Archibald C. Victor Z. Vishnu I.

October 24 2015

Instructions

Do NOT open this packet until instructed to do so. Once instructed to begin, your team will have two hours to complete this packet. This packet contains 12 questions, which are not of equal difficulty but are each worth the same amount of points. 60 points will be awarded for the first correct submission of each problem, with 5 points deducted for every prior incorrect submission of that problem. No points will be deducted for incorrect submissions if your team never produces a correct submission for that problem.

All submissions must be written in Java, and you must submit the `.java` files, not the `.class` files. Before submitting, make sure that your file contains no package declaration (it should be located in default package). All input will be from a file, and all output that you want the judges to see should be printed to the standard output channel (`System.out`). During grading, your submission will get 30 seconds of CPU time and 1 gigabyte of RAM. Good luck.

How to Read File Input

First, import the following:

```
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;
```

Then, modify the main method like so:

```
public static void main(String[] args) throws FileNotFoundException {
    Scanner input = new Scanner(new File("filename.in"));
    // solution code goes here
    input.close(); // this line is not necessary, but conventional
}
```

Use the following `Scanner` methods as appropriate:

```
int intInput = input.nextInt();           // reads next space-separated integer
double doubleInput = input.nextDouble();  // reads next space-separated double
String word = input.next();               // reads next space-separated string
String line = input.nextLine();           // reads next line of input
```

Contents

1 Space Latin

Input file: spacelatin.in

Mission Briefing

You are a new cadet at the Space Navy camp and, unfortunately, you have no idea how to speak Space Latin, the language most commonly spoken there. However, being the computer science whiz that you are, you notice a pattern in the enemy language and decide that you can, after all, find a way to communicate with the people there.

The arrangement of Space Latin words is as follows: All words are derived from English words. If the first letter of the word is a vowel (aeiouy), then the Space Latin word is exactly the same as the English word. Sometimes, however, the English word will begin with a sequence of consonants before the first vowel. In this case, the sequence of consonants should be cut from the beginning of the word, pasted at the end, and have *-oy* added to the end of the whole thing.

Input Format

The first line of the input file will contain a single integer indicating the number of sentences to follow. Each sentence will be completely in English and have a capitalized first word. Other than possibly the word "I," no other words within the sentence will be capitalized. Each sentence will end with a period. There will be no other forms of punctuation.

Output Format

Return each sentence, fully translated into Space Latin and with a capitalized first word and period at the end, on its own line.

Sample

Input

```
3
Greetings my fellow comrades.
I am a space cadet.
I love to fish in the lake.
```

Output

```
Eetingsgroy ymoy ellowfoy omradescoy.
I am a acespoy adetcoy.
I oveloy otoy ishfoy in ethoy akeloy.
```

2 Space History

Input file: history.in

Mission Briefing

The well-rounded Space Navy officer must know more than just how to speak Space Latin—he or she also needs to be familiar with space history. As such, you will be assessed on how much you know about the single most pivotal figure in the field of space travel engineering, Fizz B. Lightyear. You remember little about the man, but thank goodness for the Web, right?

The luminary F. B. Lightyear made several trips over the course of his career to test his spacecraft. In his infinite brilliance, he named his two standard models the "inFIZZity" and the "BUZZond." The "inFIZZity" is a first-class craft that will always travel a multiple of 299,792,458 meters per second. The "BUZZond" is an economy-class model that will travel for only multiples of 31,557,600 meters per second. His magnum opus, however, is the "TO INFINITY AND BEYOND" and can travel at multiples of the product of the speeds of the inFIZZity and the BUZZond. To pass the Space Navy camp's assessment, determine what model spacecraft from F. B. Lightyear travelled the given distance.

Input Format

The first line will have a single integer N signifying the number of trips that will follow where $1 \leq N \leq 99$. All subsequent lines will contain a single integer M, the number of meters F. B. Lightyear travelled on a single trip, where $1 \leq M \leq 2^{63} - 1$.

Output Format

For each trip, print "inFIZZity" if Lightyear travelled a multiple of 299,792,458 meters or "BUZZond" if he travelled a multiple of 31,557,600 meters; however, if the distance he travelled is a multiple of both 299,792,458 and 31,557,600, print "TO INFINITY AND BEYOND" instead. If the distance is not a multiple of either number, print "inferior model" alone.

Sample

Input

```
6
299792458
31557600
946728000
7494811450
9460730472580800
9999999999999999
```

Output

```
inFIZZity
BUZZond
BUZZond
inFIZZity
TO INFINITY AND BEYOND
inferior model
```

3 The Great Seal

Input file: —

Mission Briefing

Congratulations! Upon evaluation of your fluent computer science and Space Latin communication skills as well as your thorough knowledge of history, the Space Navy committee has appointed you as a new officer. They have supplied you with your own uniform and equipment, but you notice one thing is missing: the seal of the Space Navy. You ask the committee why they have not provided this with the uniform and they respond that you must print it out yourself. Your job is to output the navy seal as it appears below to the console.

Input Format

There will be no input for this problem.

Output Format

Your task is to print the following seal to the console exactly as it appears

Sample Output

```
THE GREAT SEAL
********************
********************
*****          *****
****     00     ****
***     0000     ***
**     000000     **
***     0000     ***
****     00     ****
*****          *****
********************
********************
```

4 Targeting

Input file: targeting.in

Mission Briefing

The starship U.S.S CS DEMOLITION has been completed, as far as hardware goes. However, the designers need your help implementing the targeting system software. You only have to work on the targeting system for the case when your starship will remain stationary when firing. Essentially, you need to target ships in order of a calculated *threat level* (you may assume no two enemy ships will have the same threat level) and specify the requisite *energy output* to the laser cannons in order to guarantee critical damage to the enemy ship. Further parameters will be enemy firepower, F, enemy defense rating, δ, and enemy distance from your ship, r.

$$\textit{threat level} = \frac{F^2}{r}$$

$$\textit{energy output} = (1.47)\,(r)\,(\delta)$$

Note: the distance between two points (x_1, y_1, z_1) and (x_2, y_2, z_2) is

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

Input Format

The first line will contain a single integer N, indicating the number of test cases. Then, for each test case, the first line will contain single-space-separated integer M and real numbers X, Y, Z where M is the number of enemy ships detected, $3 \leq M \leq 100000$, and X, Y, Z are the 3D real coordinates of your ship (CS DEMOLITION) in Cartesian System Coordinates. The following M lines will each correspond to a single enemy ship, containing 1 integer i corresponding to the enemy ship's unique ID and 5 real numbers x, y, z, F, δ, corresponding to the 3D Cartesian coordinates of the enemy ship, the enemy ship's FIREPOWER, and the enemy ship's DEFENSE RATING.

Output Format

For the ships with the 3 highest threat levels (from greatest threat level to least), output the enemy ship ID followed by a space and the ENERGY OUTPUT rounded to the fourth decimal place. You may assume that the energy output will be greater than .001.

Sample

Input

```
1
3 -2.7 44.92 13.1
17 2.2 3.4 -16.43 18.36 .162
42 1.9 32.6 13.0 20.17 .425
33 4.62 -11.4 31.56 15.0 .298
```

Output

```
42 8.2162
17 12.1893
33 26.1603
```

5 Crack the Code

Input file: code.in

Mission Briefing

While out on one of you guard duty days, you notice an interesting signal reverberating through the cosmos. When you track it down, you see that the signal comes from a location where intelligence suggests the enemy may be hiding. You record the signal and bring it back to the people at intelligence headquarters who, unfortunately, are on their lunch break. When you look at the content of the signal, you see that it is a series of sets of 5 numbers followed by a series of 5-word sentences. Naturally, you are confused.

You look up the data in the archives and see that such a signal has been intercepted before. All that intelligence knows is that in the set of numbers, each number corresponds to one word in the following sentence. An intuitive feeling inside you tells you that danger is approaching quickly and that you must act accordingly. It falls to you to crack this code.

Input Format

The number sequences are sent in the order “1,2,3,4,5” - “2,3,4,5,6” - “3,4,5,6,7” - “4,5,6,7,8” - “5,6,7,8,9” (note the comma in between the numbers) with each set of 5 on a successive line. After this comes a series of 5-word sentences (punctuated with a period at the end), with each of these words, as mentioned before, corresponding to a number in the set of numbers that corresponds to the sentence. For example, the first set of numbers corresponds to the words in the first sentence, although the words in the sentence may not necessarily be arranged in the same order as their corresponding numbers. Once again, each of these sentences is on its own line.

Output Format

As can be seen, there are 9 words in total that the enemy has encrypted into their numerical code. The output you wish to send to intelligence is the number 1 followed by a space and the word corresponding to it, the number 2 followed by a space and the (uncapitalized) word corresponding to it, and so on all the way to the ninth word.

Sample

Input

```
1,2,3,4,5
2,3,4,5,6
3,4,5,6,7
4,5,6,7,8
5,6,7,8,9
The man ran to him.
The man ran to her.
The woman ran to her.
A woman ran to her.
A woman ran from her.
```

Output

```
1 him
2 man
3 the
4 to
5 ran
6 her
7 woman
8 a
9 from
```

6 Navigation

Input file: navigation.in

Mission Briefing

Lt. Commander Karis Ko-ding is the proud principal navigator of the now-completed CS Demolition. Of course, no one would so much as DARE accuse her of laziness, but as it is, she finds plotting and detailing an appropriate course through the asteroid belt during secure-the-objective training a perfectly pointless, absolutely absurd, despicable, most inexcusable waste of her time. She much prefers to sleep, read sci-fi, binge-watch anime, or play Pokemon. So, in her infinite kindness and wisdom, she has delegated to *you* the most cherished responsibility—the most coveted honor, to help *her* (what a lucky soul). It's simple really. All you have to do is plot the shortest course through the asteroid field from your base that secures all 6 objectives and then takes you back to your base while, of course, avoiding crashing the ship into any asteroids. Revel in this *singular* opportunity to help the lofty, great, kind, not-lazy Lt. Commander Ko-ding.

Input Format

The first line will consist of a single integer L, $3 \leq L \leq 15$. This corresponds to a playing field that is L by L by L. Following will be L sequences of L lines each, each sequence a L by L 2-dimensional matrix of the asteroid field representing a slice in 3-dimensional space, with adjacent matrices representing adjacent slices (i.e. cumulatively this information should be stored in a 3D array or an analogous structure).

The characters used in a matrix will be limited to

- `B`, representing the base (both the starting and the ending point) (exactly 1)
- `_`, representing an open space that can be passed through
- `x`, representing a space blocked by an asteroid that cannot be passed through
- `*`, representing an objective that must be passed through (exactly 6)

Asteroids cannot be passed through. A normal open space takes 2 minutes to pass through, but an open space right next to an asteroid (but not diagonal) takes 3 minutes to pass through, since you must slow down for safety reasons (these time requirements remain consistent for spaces containing an objective) (the base takes up no time since you can't pass through it, only enter or exit it). However, you can pass through the base between objectives in order to collect them, in which case it still takes no time to pass through do to the installation of hyperluminal warp portals on its borders.

You cannot move diagonally, only move right/left, forward/backwards, or up/down. Of course, you cannot exceed the bounds of the map you have been given (rules, you know). Good luck.

Output Format

Output a single integer t, the minimum time in minutes required of a path from the base through all the objectives and back to base.

Sample

Input

```
4
__x_
B__x
x__*
x___
___*
xx__
____
_x_*
x___
_*__
_x_*
____
___x
_x*x
____
x___
```

Output

```
47
```

7 Pleasing the Captain

Input file: pleasing.in

Mission Briefing

Ensign Tomás de Hackero is tired of getting yelled at by Captain Isick Asimom. Luckily, our clever Ensign has figured out the secret behind the Captain's moods. See, the fair Captain attempts a puzzle called the Space Anagram in his copy of the *Navy Daily* each morning. If he solves it, he's in a good mood—if he doesn't, he's in a bad mood. The problem is, dear Captain Asimom is terrible at Space Anagram. Your task is simple: write a program that solves Space Anagram so that Ensign Hackero can install it on the Captain's computer so that Captain Asimom can solve Space Anagram so that Captain Asimom is in a good mood and doesn't yell at Ensign Hackero so that Ensign Hackero is in a good mood and doesn't yell at you.

Input Format

The first line will contain an integer M. The next M lines will contain a single dictionary entry in Space Gaelic, the language of Space Anagram. The line after that will contain a single integer N. The next N lines will contain a scrambled word. All dictionary entries and scrambled entries will contain only lowercase characters and no spaces.

Output Format

For each scrambled word, on one line output all words in Space Gaelic that the scrambled word could possibly be (in alphabetical order, with a single space between each word). If there are no matching words, print `unsolvable`.

Sample

Input

```
6
crioter
almeto
sugere
seltain
maersh
gersue
4
tmoeal
uergse
istalen
taintain
```

Output

```
almeto
gersue sugere
seltain
unsolvable
```

8 Minefield Navigation

Input file: minefield.in

Mission Briefing

Uh oh! It appears that the enemy has managed to mine the space lane leading to their homeworld. Fortunately, our probes were able to create a map of the minefield. Your task is to analyze the map and determine if it is possible for us to get the cruiser through the minefield, or if we will have to take a detour through the nebula of death. The directions the cruiser can move in are up, down, left, and right.

Input Format

The first line will contain an integer M, denoting the number of cases you will have to solve. For each case, you will receive a line containing 2 integers, R and C (where $2 \leq$ R & C ≤ 7500), followed by a R by C matrix of characters that represents a map of the minefield. An asterisk on the map represents a mine, and an 'o' on the map represents a clear spot big enough for the cruiser to enter.

Output Format

For each case, print "`Roll out!`" if there is a path from the top of the map to the bottom of the map. Otherwise, print "`Looks like we have to take a detour.`"

Sample

Input

```
2
5 10
oooooooo*o
*o*******o
*oo***ooo*
*oo*oo*oo*
oo**ooo*oo
13 10
****oo**oo
ooooo*oo*o
****o**oo*
**ooooo**o
o**o***ooo
oo*oo*oo**
o**ooo**oo
o**oo***o*
*o*o*o***o
*o*****o**
o*o****ooo
oo*oo*oo**
*o*ooo**oo
```

Output

```
Roll out!
Looks like we have to take a detour.
```

9 Courses

Input file: courses.in

Mission Briefing

Captain Han has a major problem. As the commander of the Luna Naval Academy, he is supposed to make sure that it is possible to meet the prerequisites for every course offered. In particular, he must make sure that there are no prerequisite loops. A prerequisite loop is when a course is its own prerequisite. (For an example, A is a prerequisite of B, which is a prerequisite of C, which is a prerequisite of A). Under normal conditions, he simply does this by hand, but this year, more pressing issues have taken up all his time. He has ordered you to write a program to perform this check for him.

Input Format

The first line will contain an integer C denoting the number of cases you will have to solve. For each case, you will receive a line containing two integers, A and B, followed by A lines containing the name of a course offered at the Luna Naval Academy and B Lines containing prerequisite relationships between two courses. The prerequisite relationships will be written in the form:

Output Format

Print ”We’ve got a problem!” if there is a prerequisite loop. Otherwise, print ”Let the cadets in!”

Sample

Input

```
2
4 4
WAR 101
ADVANCE TATICS
STRATEGIC PLANNING
LOGISTICS 101
WAR 101 is a prerequisite of ADVANCE TATICS
WAR 101 is a prerequisite of STRATEGIC PLANNING
LOGISTICS 101 is a prerequisite of ADVANCE TATICS
STRATEGIC PLANNING is a prerequisite of LOGISTICS 101
3 4
FLEET MATAINENCE
NAVAL EXPENDETURES
NAVAL ACCOUNTING
NAVAL EXPENDETURES is a prerequisite of FLEET MATAINENCE
FLEET MATAINENCE is a prerequisite of NAVAL ACCOUNTING
NAVAL EXPENDETURES is a prerequisite of NAVAL ACCOUNTING
NAVAL ACCOUNTING is a prerequisite of NAVAL EXPENDETURES
```

Output

```
Let the cadets in!
We’ve got a problem!
```

10 Naval Aviation

Input file: aviation.in

Mission Briefing

Naval Aviation is an elite rating consisting of only the best sailors on the planet. Naturally, we have a very stringent recruiting program. The cadets must take a series of written tests, followed by inspection by Admiral Shiver. In order to be accepted into Naval Aviation, you must meet a required written score, and have Admiral Shiver's approval. The process of calculating the written score is as follows: we take the raw score of each test and converting it to an adjusted score. Then, we add up all the adjusted scores to obtain the written score.

The raw score is converted to an adjusted score using this scale:

Raw:	Written:	Raw:	Written:
> 100	4.5	70-79	1.0
95-100	3.0	60-69	0.0
90-94	3.0	50-59	-0.5
80-89	2.0	< 50	-2

Input Format

The first line of the input file will contain a single integer indicating the number of cases to follow. For each case, you will be given a line containing two integers, n and m, denoting the number of tests taken and the minimum required written score respectively. The next n lines of each case will contain the name of the test the candidate took and an integer denoting the raw score earned. After all the tests, you will receive a line containing ”Shiver Approves” if the candidate is accepted by Admiral Shiver, or ”Shiver Disapproves” if the cadet is not accepted by Admiral Shiver.

Output Format

For each case, you will output a single line containing ”Welcome to Naval Aviation” if the cadet is accepted into Naval Aviation, or ”Back to ship duty” if the cadet is rejected.

Sample

Input

```
2
3 4
Phys Ed 84
Battle Training 38
Weapon safety 101
Shiver Approves
4 10
Phys Ed 84
Battle Training 38
Weapon safety 101
Advance Tatics 100
Shiver Disapproves
```

Output

```
Welcome to Naval Aviation
Back to ship duty
```