Eleventh Annual
University of Oregon Programming Competition

Saturday, February 24, 2007

Supported in part by Microsoft

Problem Contributors

David Atkins

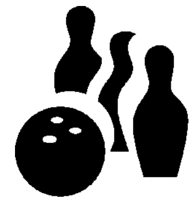
Aaron Henner

Eugene Luks

Daniel Stutzbach

Michal Young

Bowling Scores



Bowling is a game that has roots going back to ancient Egypt. In the game, players attempt to knock down pins by rolling a ball at them. The most popular form of the game in the US is known as ten-pin bowling, in which there are, not surprisingly, ten pins to knock down.

If a player knocks down all ten of the pins with one ball, that is called a strike. If all of the pins are not knocked down, then the player rolls another ball to try to knock down the pins that are still standing. If the player is successful at knocking down the rest of the pins, this is called a **spare**. After a strike or two rolls, the pins are reset, and this is called a **frame**. A game consists of ten frames.

The scoring of bowling is a rather unique system. The game score is the sum of the scores of the frames. The score for a frame is the number of pins knocked down in that frame. However, if the player had a spare in the frame, then a bonus score equal to the number of pins knocked down by the next roll (in the next frame) is added to that frame. If the player had a strike in the frame, then a bonus score equal to the number of pins knocked down by the next two rolls is added to that frame. In the tenth frame, the player may have one bonus roll if they had a spare, and two bonus rolls if they had a strike. Note that these bonus rolls do not count as an eleventh frame, but are only for the purpose of calculating the score for the tenth frame. In particular, knocking down all ten pins with one or two bonus rolls does not entitle the player to yet more bonus rolls - each bonus roll will add 0 to 10 points to the score of the tenth frame.

In this problem, you are given a sequence of games for which you are to calculate the total score. Each game will consist of a sequence of symbols indicating the number of pins knocked down by a roll. The digits 0 through 9 mean that many pins were knocked down. The symbol `X` indicates a strike, i.e., all ten pins were knocked down on that roll. The symbol / indicates a spare, i.e., all of the pins remaining standing after the previous roll were knocked down.

The first line of input will be a number n that specifies how many games are to be scored. Each of the next n lines will specify a game and consist of the symbols described above, separated by spaces.

Your output must consist of the score for each game, on a line by itself.

Sample Input

```
5
4 5 8 / 2 7 X 7 / X 1 8 8 / 7 1 3 5
4 5 8 / 2 7 X 7 / X 1 8 8 / 7 1 3 / X
4 5 8 / 2 7 X 7 / X 1 8 8 / 7 1 X X 5
4 5 8 / 2 7 X 7 / X X X 8 / 7 1
X X X X X X X X X X X X
```

Sample Output

```
131
143
148
173
300
```

WHOSE HOMEWORK

Every year Professor N.P. Hard teaches CIS 999: *3-SAT and other Intractable Problems*, and every year the professor runs into a meta-problem in dealing with homework submissions

Students are allowed to collaborate on homeworks and often find that necessary. However, they are obliged to enter the names of all their collaborators on a cover sheet downloaded from the course web page. Since collaboration is always mutual, we know that if Alice is on Bob's list then Bob will be on Alice's list.

Unfortunately, Prof. Hard neglected to leave a place for students to sign their own names and no student would be so presumptuous as to correct this professor's format by inserting an extra element in the form; as a result their own names do not appear on their homeworks.

Your job is to determine the proper owner of as many of the homeworks as possible. Fortunately, Prof. Hard's reputation always keeps the enrollment at 8 or less.

Consider the following example:

Homework 1 has listed:	*Greg*
Homework 2 has listed:	*Steve, Sarah*
Homework 3 has listed:	*Greg*

Homework 2 must belong to Greg, and that is all that can be determined.

The first line input to this problem will be an integer k indicating the number of homework sets to follow. Each homework set begins with a line of the form

$$n \ \langle\text{name } 1\rangle \ \langle\text{name } 2\rangle \ \cdots \ \langle\text{name } n\rangle$$

where n is the number of names listed on that homework. The following n lines represent the n homeworks submitted; for $1 \leq k \leq n$, the k^{th} line would be of the form

$$k \ m_k \ \langle\text{name } 1\rangle \ \langle\text{name } 2\rangle \ \cdots \ \langle\text{name } m_k\rangle$$

where m_k is the number of names listed on homework k.

The output for each homework set should indicate the homeworks that have identifiable owners, sorted by homework number (see sample output). If no owners can be determined then the output should read

`No owners can be determined.`

There should be a blank line between outputs for the successive homework sets.

Sample Input

```
3
3 Greg Steve Sarah
1 1 Greg
2 2 Steve Sarah
3 1 Greg
3 Alice Bob Eve
1 1 Bob
2 1 Alice
3 0
4 John Ringo Paul George
1 2 Ringo Paul
2 2 John George
3 2 John George
4 2 Ringo Paul
```

Sample Output

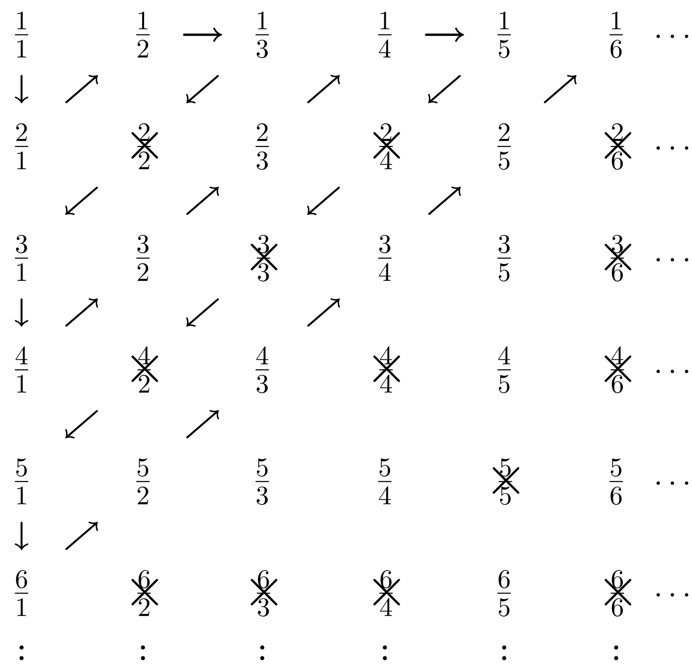
```
Homework 2 belongs to Greg

Homework 1 belongs to Alice
Homework 2 belongs to Bob
Homework 3 belongs to Eve

No owners can be determined
```

Cantor Counter

A famous proof by George Cantor [1845–1918] uses the following scheme to show that the rational numbers are countable.



Every positive rational number will eventually be reached in the sequence

$$\frac{1}{1}, \frac{2}{1}, \frac{1}{2}, \frac{1}{3}, \frac{3}{1}, \frac{4}{1}, \frac{3}{2}, \frac{2}{3}, \frac{1}{4}, \frac{1}{5}, \frac{5}{1}, \frac{6}{1}, \frac{5}{2}, \frac{4}{3}, \frac{3}{4}, \ldots$$

Notice that duplicate representations are bypassed in the enumeration so that every rational $\frac{N}{D}$ has a unique *index* in this sequence (the index of $\frac{1}{1}$ is 1, the index of $\frac{2}{1}$ is 2, the index of $\frac{1}{2}$ is 3, the index of $\frac{1}{3}$ is 4, the index of $\frac{3}{1}$ is 5, etc.) In this problem, you must determine the index of given positive rationals.

The input to this problem will be an integer n indicating the number of rationals to follow. Each of the following n lines will contain two positive integers N, D, each ≤ 1000 and separated by a single space. Note that, as input, $\frac{N}{D}$ *need not be in lowest terms.*

The output for each pair N, D should be the index of $\frac{N}{D}$ in Cantor's sequence.

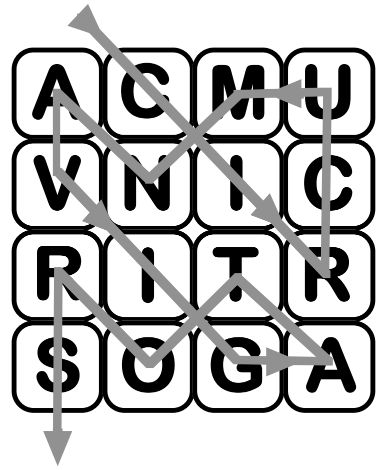
Sample Input

```
4
1 1
3 3
5 2
9 12
```

Sample Output

```
1
1
13
15
```

BOGGLEMATIC



Boggle is a word game designed by Allan Turoff and trademarked by Parker Brothers. 16 dice, each die with a letter* on each face, are shaken and placed on a 4×4 game board.

Each player tries to form as many words as possible (three letters or longer, excluding proper nouns) in a three minute period. A word on a Boggle board is constructed by starting at any square in the grid and proceeding sequentially along a path. Each step in the path may be one step in a horizontal, vertical, or diagonal direction, and must not be to a square that has already been used.

Being programmers, we soon get tired of thinking, and instead set out to automate the game.

For purposes of the contest, instead of finding as many words as possible, the goal is to find one word that is as long as possible. Words are provided in a word list, so all we have to do is pick the longest word from the list that can be spelled out on a given boggle board.

The word list is a file named "`dict`", which has already been placed in your home directory. Each line of "`dict`"' contains one word consisting only of lower case letters ('`a`' through '`z`') and the list appears in alphabetic order. No word is longer than 16 letters, and the list of words is no longer than 100,000 words. A line containing only a period (".") marks the end of the word list.

The first line of the program input is a positive integer n. It is followed by n Boggle boards. A Boggle board is 4 lines of 4 alphabetic characters each, followed by a line containing only a period (".")

Your program should print one line for each Boggle board in the input. If at least one word from the word list of at least three characters can be constructed from the Boggle board, the program prints the longest such word. If two or more words are tied for longest, then the word that appears first in alphabetical order is selected for printing. If none of the words in the word list can be constructed on the Boggle board, the program prints

```
Rats! We lose!
```

*Actually, `Q` and `U` appear together on one face, but for the purposes of this problem we will not treat `Q` specially.

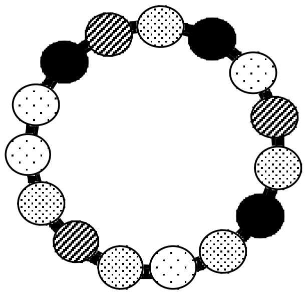
Sample Input

```
5
acmu
vnic
ritr
soga
.
uhob
ceod
lsal
evif
.
hpai
oylo
gqmn
efen
.
xbns
xqqr
wcdx
flzo
.
eess
rnia
swth
nhtx
.
```

Sample Output

```
circumnavigators
cleaves
laymen
Rats! We lose!
wieners
```

COLORED MARBLES



While shopping in an art museum store, Dr. Dan came across a game to help teach children about primary and secondary pigments. The game consisted of a circular groove, filled with marbles. The marbles may be a primary pigment (red, blue, or yellow) or a secondary pigment (green, orange, or purple).

Players may remove a pair of adjacent marbles only if one is a primary pigment, and the other is a secondary pigment based on that primary pigment. In other words, the following pairs may be removed:

- red–orange
- red–purple
- blue–green
- blue–purple
- yellow–green
- yellow–orange

Players take turns removing pairs of marbles. If a player cannot remove any marbles, he or she loses the game. Therefore, the object of the game is to force such a situation. Dr. Dan would like to be able to predict which player will win the game, assuming that there are two players and both play optimally. You are to write a program that determines which player will win.

The first line of input will be a number n that specifies how many games will follow. Each of the next n lines will specify a game as a string, with each character of the string representing one marble. Two adjacent characters represent adjacent marbles on the board. Since the board is circular, the first character and last character also represent adjacent marbles. For example, the line

```
rbygop
```

represents a board with 6 marbles in the following order: red, blue, yellow, green, orange, and purple. The yellow and green marbles may be removed, as may the red and purple marbles, leaving the blue and orange marbles which may not be removed.

Each game may have up to 25 marbles.

The output for each game should indicate whether Player 1 or Player 2 wins, using the format demonstrated below.

Sample Input

```
3
rbygop
rbyrbyrby
rbprgr
```

Sample Output

```
Player 2 wins.
Player 2 wins.
Player 1 wins.
```

Explanation.

If the first game, Player 1 must remove either the red and purples marbles or the yellow and green marbles. In either case, Player 2 removes the other pair, leaving Player 1 with no valid moves. Player 2 wins.

In the second game, there are no valid moves to begin with, so Player 1 loses and Player 2 wins.

In the third game, Player 1 can force a win by removing the blue and purple marbles, leaving Player 2 with no valid moves. If Player 1 instead removed the purple and red marbles, this would allow Player 2 to win and we have assumed that both players are playing optimally.