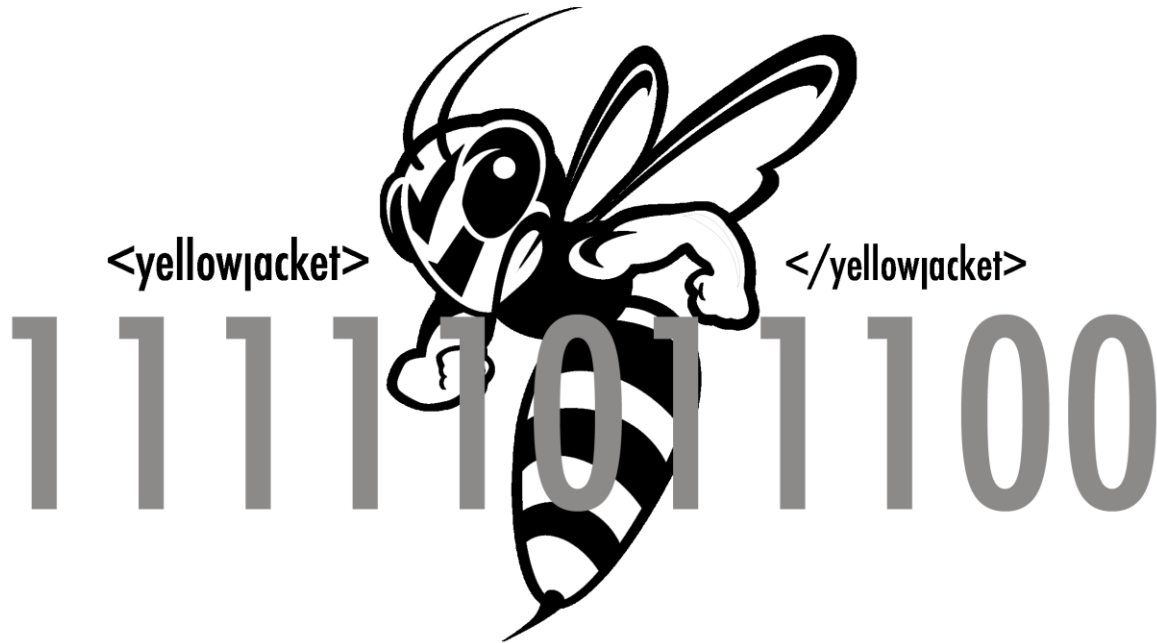


BALDWIN-WALLACE COLLEGE  
2012 PROGRAMMING CONTEST



**DO NOT OPEN UNTIL INSTRUCTED!!!!**



## *Walkin' the Dog*

Bob likes to walk his dog in unfamiliar neighborhoods. He begins his walk at a 4-way intersection by going left (L), right (R), or straight (S). Bob doesn't know the neighborhood too well, so he decides to record the direction (L, R, or S) he takes at each intersection that he and his dog encounter. Eventually, Bob will make a U-turn and use his written directions to retrace his path back to the starting point.

You are to write the directions he should follow (R, L, S), in order to retrace his path, beginning at the U-turn and returning to his starting point.

### **Details of the Input**

The first line of input will contain a single integer,  $n$  ( $0 < n < 200$ ), indicating the number of walks that Bob took. The next  $n$  lines each specify the path Bob took to the U-turn on a single walk. A path will be given as a sequence of the characters L, R, and S, terminating with a U. Each walk will be between 3 and 200 characters long.

### **Details of the Output**

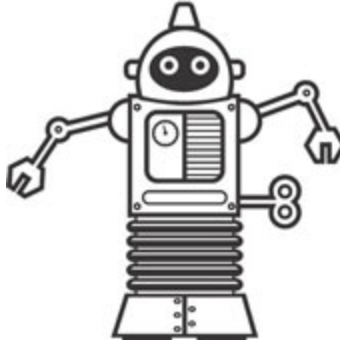
Output for each of the  $n$  walks should be on a separate line. Begin each line of output with "Walk  $k$ :", using the appropriate walk number  $k$ , followed by the path Bob must take in order to return to the starting position. It should be given in terms of the directions (R, L, S) to be followed at each intersection in order to return to the starting position.

### **Sample Input**

```
3
SRLRLU
SSLRRRLSLRLU
SSSSRSLLSRU
```

### **Sample Output**

```
Walk 1: SRLRRL
Walk 2: SRLRSRLLLRSL
Walk 3: SLSRRSLSSSS
```



## “Basic” Baskets

You've been taking a rather challenging introductory basket weaving course this semester. Some of your fellow students have been getting unusually high scores on the math sections of the basket weaving exams. Unfortunately, you are not one of those students. After some intense spying, you discover that many of them have been visiting Robbie to get homework answers. You decide to meet this Robbie for yourself, but learn that he's a robot and can't speak English! He can only communicate homework answers in unusual number base formats. In order to prove your classmates are cheating, you need to be able to decode the answers he is providing.

Studying the answers Robbie gives, you have discovered several things. The answer to each question in a given homework assignment is written in a different base. For each number, its base is the lowest base in which the given representation is valid. Although the order in which Robbie answers the questions for a given homework assignment is random, they'd be in the proper order if you consider them in the order of their bases. That is, the bases in which the answers are given increase from the first answer to the last. Oh ... why does Robbie have to make things so difficult!

### Details of the input

Input will consist of several homework assignments. The first line of each homework assignment will be a single integer  $n$  that indicates the number of answers to follow. The next  $n$  lines contain answers for that homework assignment, one per line. A value of 0 for  $n$  indicates the end of input. An individual answer is a positive integer expressed in some base  $b$  ( $2 \leq b \leq 21$ ). No two answers on the same homework assignment have the same base.

### Details of the output

Output for each homework assignment should begin with the line “HOMEWORK  $k$ ” where  $k$  is the appropriate homework number. This should be followed by the  $n$  answers to the  $n$  questions on that homework assignment, one per line, in base ten. Each answer should be preceded by the number of the problem to which it is the answer (1. –  $n$ ). A blank line should follow the listing for each homework assignment.

### Sample Input

3  
2B5  
1525  
AC38  
5  
8GH  
14  
B08  
1337  
816D  
0

### Sample Output

HOMEWORK 1

1. 413
2. 425
3. 24045

HOMEWORK 2

1. 9
2. 735
3. 1592
4. 22245
5. 2897



## Bingo Board Correction

You and your friend, Jack, are planning to host a Bingo game at a local church for a charity event. The Bingo boards are supposed to have 25 numbers on them, arranged in 5 columns of 5 numbers each, and the columns identified by the letters B, I, N, G, and O respectively. Numbers 1 – 5 (in that order) are to form the first column, 6 – 10, the second, etc. Jack has printed out all of the boards for the event, but you notice there is an error. When all of the boards were printed, several numbers were missing. Jack has a program that will print numbers in a specified row and column, but needs to find the missing numbers manually. He knows that you are a skilled programmer, and asks you to create a program to find the missing numbers for him.

### Details of the Input

The first line of input will contain a single integer,  $n$ , indicating the number of Bingo boards to follow. Each Bingo board will be specified by five lines corresponding to the five rows of five numbers each. The numbers for each row are separated by a single space. If a number is missing, a 0 appears in its place. For example, the row 1 6 0 16 0 is the first row of a Bingo board and has the third and fifth numbers missing.

### Details of the Output

For each of the Bingo boards, begin its relevant output with the line “Board  $k$ ”, where  $k$  is the board number to which the lines immediately following refer. For each board, you are to identify where the missing numbers are, and what the missing value is. Information about each missing value should be on a separate line, and should include the column (B, I, N, G, or O), the row number (1 – 5), and the missing value, in that order. These should be separated by a single space. For example, a number missing in column I, row 2, would output “I 2 7” where 7 is the number that should be in that board position. The missing number information should identify any missing numbers in row one before any in row two, etc. Within a row, missing numbers should be identified from left to right. A blank line should follow the output for each Board.

### Sample Input

```
3
1 6 0 16 0
2 7 12 17 22
0 8 13 0 23
4 9 0 0 0
5 10 15 20 25
0 6 11 0 21
2 7 12 0 22
3 8 0 18 23
0 9 14 19 24
5 10 0 20 0
1 6 11 16 21
2 0 12 0 0
3 8 13 0 23
4 0 14 0 24
0 0 0 0 0
```

### Sample Output

Board 1

```
N 1 11
O 1 21
B 3 3
G 3 18
N 4 14
G 4 19
O 4 24
```

Board 2

```
B 1 1
G 1 16
G 2 17
N 3 13
B 4 4
O 5 25
```

Board 3

```
I 2 7
G 2 17
O 2 22
G 3 18
I 4 9
G 4 19
B 5 5
I 5 10
N 5 15
G 5 20
O 5 25
```



## The Legend of Neil and the Turning Tower

While exploring the fourth floor of an old, rundown tower, Neil got lost and was unsure how to get out. The tower is simply shaped, with each of the levels having the same dimensions and layout. The floor on each level is constructed of square tiles in a  $6 \times 6$  grid, but some of the tiles are missing. If Neil isn't careful, he will fall through a hole resulting from a missing tile and end up gravely injured on the floor below, or be lucky enough to fall through to the basement, that is full of water, and hence will be saved.

As Neil looks around the room on the fourth floor, he sees 3 levers on the wall labeled A, B, and C. He hesitates, but then pulls C, the 3<sup>rd</sup> lever. From one of the holes in his floor, he sees the floor directly beneath his rotating clockwise about the center of the room. It rotated 90 degrees! Seeing this, Neil realizes that one of the empty tile spaces on the third floor aligned with the hole he is looking through on the fourth floor!

That gave Neil an idea! Maybe he can rotate the floors below in such a way as to align the holes. After a few tries, Neil has discovered that lever A rotates the first floor, B, the second floor, and C, the third floor. If he can only align holes in the three floors below with a hole on the fourth floor, he can jump in and fall straight through all the floors, landing safely in the water in the basement.

### Details of the Input

Luckily, Neil lives in a world where he has several lives, up to 99 to be precise, so he can make multiple attempts at getting out. And in Neil's world, as each of his lives begins, he finds himself back on the fourth floor of a different, but similarly functioning tower. The first line of the input will contain the single positive integer  $n$  ( $0 < n < 100$ ), representing the number of towers he will attempt to exit.

Each attempt will be indicated by 25 lines of input. The first 24 of those (four sets of six) represent the four floors of the tower from the bottom up (that is, first through fourth). Each floor representation contains 6 values in the range 0 through 2. A 1 represents solid floor while a 0 represents a hole in the tiles through which Neil can jump. The value 2 marks the location of the hole in the fourth floor through which Neil jumps. The 25<sup>th</sup> line contains an integer and a string separated by a single blank space. The integer,  $p$  ( $0 < p \leq 25$ ), indicates the number of lever-pulls

Neil makes. The string contains  $p$  characters in the range A through C, each indicating a specific levers Neil pulls.

### Details of the Output

There should be a single line of output for each of the  $n$  attempts. It should begin with “Attempt  $k$ :”, where  $k$  is the appropriate attempt number, followed by a blank and the word “Lived” if Neil made it through each floor or “Died” if he hit a tile without a hole on the way down.

### Sample Input

```
1
110110
011101
111111
111011
101010
111011
110110
011101
111111
111011
101010
111011
110110
011101
111111
111011
101010
111011
110110
011101
111111
111011
101010
111211
10 AAAABBABCC
```

### Sample Output

```
Attempt 1: Died
```





## Go deliver a dare, vile dog

You wish to write for your college's newspaper, but find out there is a sort of qualifying test. Anyone wishing to write for the paper must first demonstrate a particular language skill -- being able to identify palindromes. From your previous English classes, you know that a sentence or word is a palindrome if, spelled backwards, it is the same as the original. You will be asked to recognize palindromes from a list of words and phrases.

### Details of the Input

The first line of input will contain a single positive integer,  $n$ , indicating the number of lines to follow, each of which is a string to be tested. Although these strings may contain punctuation and/or spaces, you are to ignore those while making your determination. Also disregard capitalization.

### Details of the Output

There should be one line of output for each string to be tested. It should begin with the case number and a colon, followed by the appropriate phrase, either *palindrome* or *not a palindrome*.

### Sample Input

```
3
tacocat
hello
A Toyota.
```

### Sample Output

```
Case 1: palindrome
Case 2: not a palindrome
Case 3: palindrome
```



## The Most Advanced Calculator OF ALL TIME

For some unknown reason, your teacher wants you to make a calculator that accepts English representations of integers and operators and outputs the results of the operations. From past experience, you have learned to not ask questions ... just do it.

Your program is to run on a specific set of operands and operators (see full list below). It is to perform the specified operation on the indicated operands, if possible, and write the results. It should also indicate when it is not possible.

### Possible operands

zero	five
one	six
two	seven
three	eight
four	nine

### Possible operators

plus  
minus  
times  
div

### Details of the Input

The input will begin with a positive integer  $n$ , signifying the number of problems to follow. Each of the following  $n$  lines will contain three words (corresponding to *operand1*, *operator*, and *operand2*, respectively) separated by a single space.

### Details of the Output

Your program should produce one line for each of the  $n$  problems. If the problem is valid, your program should write the entire operation, including the answer, in numeric form. All operations should be considered integer operations. A problem is valid only if the operation and both operands are among the listed possibilities, and the mathematical operation is valid. If the problem is not valid, your program should write “fail”.

### Sample input

```
5
one plus two
four times five
nine div five
one div three
1million squared 2
```

### Sample output

```
1 + 2 = 3
4 * 5 = 20
9 / 5 = 1
1 / 3 = 0
fail
```