Problem (A)

### 673 - Parentheses Balance

You are given a string consisting of parentheses () and []. A string of this type is said to be *correct*:

(a) if it is the empty string

(b) if A and B are correct, AB is correct,

(c) if A is correct, (A) and [A] is correct.

Write a program that takes a sequence of strings of this type and check their correctness. Your program can assume that the maximum string length is 128.

**Input**

The file contains a positive integer *n* and a sequence of *n* strings of parentheses () and [], one string a line.

**Output**

A sequence of Yes or No on the output file.

**Sample Input**

3

([])

(([()])))

([()[]()])()

**Sample Output**

Yes

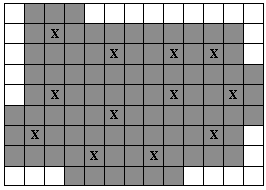
No

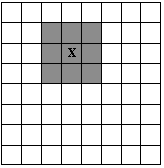
Yes

Problem (B)

### 11044 - Searching for Nessy

In July 2003, the BBC reported an extensive investigation of Loch Ness by a BBC team, using 600 separate sonar beams, found no trace of any ¨sea monster¨ (i.e., any large animal, known or unknown) in the loch. The BBC team concluded that Nessie does not exist. Now we want to repeat the experiment. Given a grid of ***n*** rows and ***m*** columns representing the loch, 6$ \le$***n***, ***m***$ \le$10000, find the minimum number ***s*** of sonar beams you must put in the square such that we can control every position in the grid, with the following conditions:

* one sonar occupies one position in the grid; the sonar beam controls its own cell and the contiguous cells;
* the border cells do not need to be controlled, because Nessy cannot hide there (she is too big).

For example,

where ***X***  represents a sonar, and the shaded cells are controlled by their sonar beams; the last figure gives us a solution.

**Input**

The first line of the input contains an integer, ***t***, indicating the number of test cases. For each test case, there is a line with two numbers separated by blanks, 6$ \le$***n***, ***m***$ \le$10000, that is, the size of the grid (***n*** rows and ***m*** columns).

**Output**

For each test case, the output should consist of one line showing the minimum number of sonars that verifies the conditions above.

**Sample Input**

**Sample Output**

4

4

12

3

6 6

7 7

9 13

# Problem (C)

### 12554 - A Special "Happy Birthday" Song!!!

There are *n* people (excluding myself) in my 30th birthday party. They sing the traditional "happy birthday" song:

Happy birthday to you! Happy birthday to you! Happy birthday to Rujia! Happy birthday to you!!!

Since I love music, I want to hear something more interesting, not that everyone sings together. Ah yes, I want one person to sing one word!

For example, there are three people: Mom, Dad, Girlfriend, I'd like them to sing like this:

Mom: Happy

Dad: birthday

Girlfriend: to

Mom: you

Dad: Happy

Girlfriend: birthday

Mom: to

Dad: you

Girlfriend: Happy

Mom: birthday

Dad: to

Girlfriend: Rujia

Mom: Happy

Dad: birthday

Girlfriend: to

Mom: you

Very nice, right? What if there are more than 16 people? That's easy: repeat the song until everyone has sung at least once :)

Please, don't stop in the middle of the song.

**Input**

There is only one test case. The first line contains a single integer *n* (1<=*n*<=100). Then each of the next n lines contains a capitalized name (i.e. one upper-case letter followed by zero or more lower-case letters). Each name contains at most 100 characters and do not have whitespace characters inside.

**Output for the Sample Input**

Mom: Happy

Dad: birthday

Girlfriend: to

Mom: you

Dad: Happy

Girlfriend: birthday

Mom: to

Dad: you

Girlfriend: Happy

Mom: birthday

Dad: to

Girlfriend: Rujia

Mom: Happy

Dad: birthday

Girlfriend: to

Mom: you

**Output**

Output the song, formatted as above.

**Sample Input**

3

Mom

Dad

Girlfriend

Problem (D)

### 11616 - Roman Numerals

**The Problem**

A Roman numeral consists of a set of letters of the alphabet. Each letter has a particular value, as shown in the following table:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Letter | I | V | X | L | C | D | M |
| Value | 1 | 5 | 10 | 50 | 100 | 500 | 1000 |

Generally, Roman numerals are written in descending order from left to right, and are added sequentially. However, certain combinations employ a subtractive principle. If a symbol of smaller value precedes a symbol of larger value, the smaller value is subtracted from the larger value, and the result is added to the total. This subtractive principle follows the next rules:

* "I" may only precede "V" and "X" (e.g., IV=4).
* "X" may only precede "L" and "C" (e.g., XC=900).
* "C" may only precede "D" and "M".
* "V", "L" and "D" are always followed by a symbol of smaller value, so they are always added to the total.

Symbols "I", "X", "C" and "M" cannot appear more than three consecutive times. Symbols "V", "L" and "D" cannot appear more than once consecutively.

Roman numerals do not include the number zero, and for values greater or equal than 4000 they used bars placed above the letters to indicate multiplication by 1000.

You have write a program that converts from Roman to Arabic numerals and vice versa. Although lower case letters were used in the Middle Ages, the Romans only used upper case letters. Therefore, for the Roman numerals we only consider upper case letters.

**The Input**

The input consists of several lines, each one containing either an Arabic or a Roman number *n*, where 0 < *n* < 4000.

**The Output**

For each input line, you must print a line with the converted number. If the number is Arabic, you must give it in Roman format. If the number is Roman, you must give it in Arabic format.

**Sample Input**

**Sample Output**

25

IV

CMXLII

1983

XXV

4

942

MCMLXXXIII

# Problem (E)

### 11849 – CD

Jack and Jill have decided to sell some of their Compact Discs, while they still have some value. They have decided to sell one of each of the CD titles that they both own. How many CDs can Jack and Jill sell?

Neither Jack nor Jill owns more than one copy of each CD.

### Input Specification

The input consists of a sequence of test cases. The first line of each test case contains two non-negative integers N and M, each at most one million, specifying the number of CDs owned by Jack and by Jill, respectively. This line is followed by N lines listing the catalog numbers of the CDs owned by Jack in increasing order, and M more lines listing the catalog numbers of the CDs owned by Jill in increasing order. Each catalog number is a positive integer no greater than one billion. The input is terminated by a line containing two zeros. This last line is not a test case and should not be processed.

### Sample Input

3 3

1

2

3

1

2

4

0 0

### Output Specification

For each test case, output a line containing one integer, the number of CDs that Jack and Jill both own.

### Output for Sample Input

2

Problem (F)

### 10921 - Find the Telephone

In some places is common to remember a phone number associating its digits to letters. In this way the expression **MY LOVE** means **69 5683**. Of course there are some problems, because some phone numbers can not form a word or a phrase and the digits **1** and **0** are not associated to any letter.

Your task is to read an expression and find the corresponding phone number based on the table below. An expression is composed by the capital letters (**A-Z**), hyphens (**-**) and the numbers **1** and **0**.

|  |  |
| --- | --- |
| Letters | Number |
| ABC | 2 |
| DEF | 3 |
| GHI | 4 |
| JKL | 5 |
| MNO | 6 |
| PQRS | 7 |
| TUV | 8 |
| WXYZ | 9 |

**Input**

The input consists of a set of expressions. Each expression is in a line by itself and has **C** characters, where **1 ≤ C ≤ 30**. The input is terminated by enf of file (EOF).

**Output**

For each expression you should print the corresponding phone number.

**Sample Input**

1-HOME-SWEET-HOME

MY-MISERABLE-JOB

**Sample Output**

1-4663-79338-4663

69-647372253-562

# Problem (G)

### 401 - Palindromes

A regular palindrome is a string of numbers or letters that is the same forward as backward. For example, the string "ABCDEDCBA" is a palindrome because it is the same when the string is read from left to right as when the string is read from right to left.  
A mirrored string is a string for which when each of the elements of the string is changed to its reverse (if it has a reverse) and the string is read backwards the result is the same as the original string. For example, the string "3AIAE" is a mirrored string because "A" and "I" are their own reverses, and "3" and "E" are each others' reverses.

A mirrored palindrome is a string that meets the criteria of a regular palindrome and the criteria of a mirrored string. The string "ATOYOTA" is a mirrored palindrome because if the string is read backwards, the string is the same as the original and because if each of the characters is replaced by its reverse and the result is read backwards, the result is the same as the original string. Of course, "A", "T", "O", and "Y" are all their own reverses.

A list of all valid characters and their reverses is as follows.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Character | Reverse | Character | Reverse | Character | Reverse |
| A | A | M | M | Y | Y |
| B |  | N |  | Z | 5 |
| C |  | O | O | 1 | 1 |
| D |  | P |  | 2 | S |
| E | 3 | Q |  | 3 | E |
| F |  | R |  | 4 |  |
| G |  | S | 2 | 5 | Z |
| H | H | T | T | 6 |  |
| I | I | U | U | 7 |  |
| J | L | V | V | 8 | 8 |
| K |  | W | W | 9 |  |
| L | J | X | X |  |  |

**Note** that O (zero) and 0 (the letter) are considered the same character and therefore **ONLY** the letter "0" is a valid character.

**Input**

Input consists of strings (one per line) each of which will consist of one to twenty valid characters. There will be no invalid characters in any of the strings. Your program should read to the end of file.

**Output**

For each input string, you should print the string starting in column 1 immediately followed by exactly one of the following strings.

|  |  |
| --- | --- |
| STRING | CRITERIA |
| " -- is not a palindrome." | if the string is not a palindrome and is not a mirrored string |
| " -- is a regular palindrome." | if the string is a palindrome and is not a mirrored string |
| " -- is a mirrored string." | if the string is not a palindrome and is a mirrored string |
| " -- is a mirrored palindrome." | if the string is a palindrome and is a mirrored string |

**Note** that the output line is to include the -'s and spacing exactly as shown in the table above and demonstrated in the Sample Output below.

In addition, after each output line, you must print an empty line.

**Sample Input**

NOTAPALINDROME

ISAPALINILAPASI

2A3MEAS

ATOYOTA

**Sample Output**

NOTAPALINDROME -- is not a palindrome.

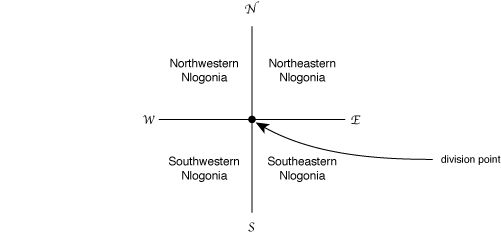
ISAPALINILAPASI -- is a regular palindrome.

2A3MEAS -- is a mirrored string.

ATOYOTA -- is a mirrored palindrome.

# Problem (H)

# 11498 - Division of Nlogonia

Nlogonia was divided into four independent territories ,using *division point*, in the following way. Two lines, both containing the division point, one in the North-South direction and one in the East-West direction, would be drawn on the map, dividing the land into four new countries. Starting from the Western-most, Northern-most quadrant, in clockwise direction, the new countries will be called Northwestern Nlogonia, Northeastern Nlogonia, Southeastern Nlogonia and Southwestern Nlogonia.

**The Input**

The input contains several test cases. The first line of a test case contains one integer *K* indicating the number of queries that will be made (*0 < K ≤ 103*). The second line of a test case contains two integers *N* and *M* representing the coordinates of the division point (*-104 < N, M < 104*). Each of the *K* following lines contains two integers *X* and *Y* representing the coordinates of a residence (*-104 ≤ X, Y ≤ 104*).

The end of input is indicated by a line containing only the number zero.

**The Output**

For each test case in the input your program must print one line containing:the word divisa (means border in Portuguese) if the residence is on one of the border lines (North-South or East-West);

* NO (means NW in Portuguese) if the residence is in Northwestern Nlogonia;
* NE if the residence is in Northeastern Nlogonia;
* SE if the residence is in Southeastern Nlogonia;
* SO (means SW in Portuguese) if the residence is in Southwestern Nlogonia.

**Sample Input**

**Sample Output**

NE

divisa

NO

divisa

NE

SO

SE

3

2 1

10 10

-10 1

0 33

4

-1000 -1000

-1000 -1000

0 0

-2000 -10000

-999 -1001

0

# Problem (I)

### 484 - The Department of Redundancy Department

Write a program that will remove all duplicates from a sequence of integers and print the list of unique integers occuring in the input sequence, along with the number of occurences of each.

**Input**

The input file will contain a sequence of integers (positive, negative, and/or zero). The input file may be arbitrarily long.

**Output**

The output for this program will be a sequence of ordered pairs, separated by newlines. The first element of the pair must be an integer from the input file. The second element must be the number of times that that particular integer appeared in the input file. The elements in each pair are to be separated by space characters. The integers are to appear in the order in which they were contained in the input file.

**Sample Input**

3 1 2 2 1 3 5 3 3 2

**Sample Output**

3 4

1 2

2 3

5 1

Problem (J)

6772. Happy Coins (HC) SPOJ

A line of coins are given, some belongs to lxh while others belongs to hhb. Now the two guys are about to play a game.

In every round of game, each player can select any two consecutive coins and change them into one coin, if the two coins belongs to the same person, the new one will be gained to hhb, or else it will belongs to lxh.

The two guys take turns playing this game, lxh always play first.

The game runs round by round.

You can easily make sense that we will get only one coin at the end of game. Now the question is, suppose the two players perform optimally in the game, which person does the final coin belongs to?

### Input

The first line of the input contains one integer *T*, which indicate the number of test cases.

Following each test, the first line contains an integer *N* (*N* ≤ 105), the number of coins. Following *N* lines, each line contains a name, lxh or hhb, the name of whom this coins belongs to.

### Output

For each case, output a line contains the name of whom the final coin belongs to.

### Sample Input

1  
2  
lxh  
hhb

### Sample Output

lxh