2008 Consortium for Computing Sciences in Colleges
Programming Contest
Saturday, November 8th
Augusta State University
Augusta, GA

There are seven (7) problems in this packet. Each team member should have a copy of the problems. These problems are NOT necessarily sorted by difficulty. You may solve them in any order. Each team will receive a small prize in the color indicated below upon solving a problem correctly.

Remember input/output for the contest will be from `stdin` to `stdout`. `stderr` will be ignored. Do not refer to or use external files in your source code.

Have Fun & Good Luck! ☺

Problem 1. Recount *(pink prize)*

Problem 2. Roman Numerals *(aqua prize)*

Problem 3. Disney Movie Vault *(light pink prize)*

Problem 4. The Finnish Chef *(peach prize)*

Problem 5. The Election *(emerald green prize)*

Problem 6. Great Caesar Cipher *(red prize)*

Problem 7. Stacks of Money *(yellow prize)*

Problem 1

RECOUNT

Hopefully, the results of the 2008 presidential election have now been finalized. But eight years ago on this day, they were not. In 2000, it was “Florida, Florida, Florida” that we ended up waiting on until well into December. The “hanging chad” election between Bush v. Gore was finally resolved after an exhaustive hand recount in Florida, and the Supreme Court being called upon to put an end to the circus atmosphere there.

Your job is to read in the results from one past presidential election from the last forty years (1968 – 2008) involving two presidential candidates. You will read in the results state by state, recount the total electoral votes earned by each candidate, and print the winner. We guarantee the data will result in a single winner. The winner is the candidate with the most electoral votes, and it takes 270 electoral votes to win.

Input
The input will consist of 51 lines, each line representing the results from one of the fifty states or the District of Columbia. Each line of input will consist of three tokens in the format below: a state, electoral vote count, and the last name of the winner. You may assume that neither the state string nor the candidate string will contain any blanks, and that the electoral vote count is a valid 32-bit integer. There will always be exactly two presidential candidates.

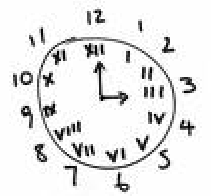
Output
Output the name of the winner and their total electoral votes in the format below.

Sample Input
```
AL 9 Bush
AK 3 Bush
AZ 8 Bush
AR 6 Bush
CA 54 Gore
CO 8 Bush
CT 8 Gore
DE 3 Gore
DC 3 Gore
FL 25 Bush
  .
  .
  .
WV 5 Bush
WI 11 Gore
WY 3 Bush
```

Output Corresponding to Sample Input
```
Bush wins with 271 votes
```

Problem 2

Roman Numerals



The Romans used letters from their Latin alphabet to represent each of the seven numerals in their number system. The list below shows which letters they used and what numeric value each of those letters represents:

I = 1
V = 5
X = 10
L = 50
C = 100
D = 500
M = 1000

Using these seven numerals, any desired number can be formed by following the two basic *additive* and *subtractive* rules. To form a number using the additive rule the Roman numerals are simply written from left to right in descending order, and the value of each roman numeral is added together. For example, the number MMCLVII has the value 1000 + 1000 + 100 + 50 + 5 + 1 + 1 = 2157. Using the addition rule alone could lead to very long strings of letters, so the subtraction rule was invented as a result. Using this rule, a smaller Roman numeral to the left of a larger one is subtracted from the total. In other words, the number MCMXIV is interpreted as 1000 − 100 + 1000 + 10 − 1 + 5 = 1914.

Over time the Roman numerals became more standardized and several additional rules were developed. The additional rules used today are:

1. The I, X, or C Roman numerals may only be repeated up to three times in succession. In other words, the number 4 must be represented as IV and *not* as IIII.
2. The V, L, or D numerals may never be repeated in succession, and the M numeral may be repeated as many times as necessary.
3. Only one smaller numeral can be placed to the left of another. For example, the number 18 is represented as XVIII but *not* as XIIX.
4. Only the I, X, or C can be used as subtractive numerals.
5. A subtractive I can only be used to the left of a V or X. Likewise a X can only appear to the left of a L or C, and a C can only be used to the left of a D or M. For example, 49 must be written as XLIX and *not* as IL.

Your goal is to write a program which converts Roman numerals to base 10 integers.

Input

The input to this problem will consist of the following:

- A line with a single integer N ($1 \leq N \leq 1000$), where N indicates how many Roman numerals are to be converted.
- A series of N lines of input with each line containing one Roman numeral. Each Roman numeral will be in the range of 1 to 10,000 (inclusive) and will obey all of the rules laid out in the problem's introduction.

Output
For each of the N Roman numerals, print the equivalent base 10 integer, one per line.

Sample Input
```
3
IX
MMDCII
DXII
```

Output Corresponding to Sample Input
```
9
2602
512
```

Problem 3

Disney Movie Vault

You have a big Disney DVD collection in your closet that you are interested in arranging in order from the most recent one to the oldest one. Your job is to write a program which takes as input a listing of movie titles, and the year they were released. Your program should provide an output listing by title arranged in descending order by year. Movies from the same year within this listing should be grouped in ascending order by title. Any movie titles that begin with the articles “A”, “An”, or “The” should move this article to the end of the title preceded by a comma & space (just as your DVR would do).

Input

Input will consist of one or more lines of input each representing a movie. The start of the line will consist of a four digit year. This will be followed by a single space and the movie title. Input will be terminated by the end of file. You may assume all articles appear exactly as “`A`”, “`An`”, or “`The`”, and appear at the start of the string followed by a single space and at least one word which is not an article. You would never see an article written as “`tHe`” or “`aN`”. Be sure any movie titles containing articles are modified before any sorting on them is done. No movie title will be longer than 75 characters.

Output

Your output should contain the same number of lines as your input. Each line in the output should start with and consist of movie titles only, modified as described.

Sample Input

```
1981 The Fox and the Hound
1998 A Bug's Life
1940 Pinocchio
1940 Fantasia
1950 Cinderella
1950 Alice in Wonderland
1950 Peter Pan
2008 WALL-E
1959 Sleeping Beauty
1967 The Jungle Book
1996 The Hunchback of Notre Dame
1977 The Rescuers
2007 Ratatouille
1982 Tron
1989 The Little Mermaid
1991 Beauty and the Beast
1994 The Lion King
1992 Aladdin
1995 Toy Story
1999 Tarzan
2003 Brother Bear
2006 Cars
```

Output Corresponding to Sample Input

```
WALL-E
Ratatouille
Cars
Brother Bear
Tarzan
Bug's Life, A
Hunchback of Notre Dame, The
Toy Story
Lion King, The
Aladdin
Beauty and the Beast
Little Mermaid, The
Tron
Fox and the Hound, The
Rescuers, The
Jungle Book, The
Sleeping Beauty
Alice in Wonderland
Cinderella
Peter Pan
Fantasia
Pinocchio
```

Problem 4

The Finnish Chef

The Muppet Show is coming back to television this Thanksgiving weekend, and you have been hired to help write scripts for the next episode. One of the main characters on the Muppet Show used to be the Swedish Chef, but he has retired. He has a younger cousin, the Finnish Chef, who will take over for him in the kitchen.

To add a Finnish flair to our recipes, you will need to be able to translate English into Mock Finnish. The Mock Finnish language has some features of the real Finnish language. Accomplish the English-to-Mock Finnish translation in these two steps.

Step #1: There are five prepositions that we are going to transform. Each time you encounter one of these prepositions, you need to delete the preposition, and then add a suffix to the first two words after the preposition. The appropriate suffix is selected from the following table. However, if either of these next two words is “a” or “the”, do not modify it. You may assume that neither of these next two words is another preposition. Assume that vowels are the letters ‘a’, ‘e’, ‘i', ‘o’, ‘u’ and ‘y’.

Preposition	Suffix to add if the word ends in a consonant:	Suffix to add if the word ends in a vowel:
`in`	`“issa”`	`“ssa”`
`from`	`“ista”`	`“sta”`
`into`	`“oon”`	Duplicate the vowel and add `“n”`. For example `“spice”` becomes `“spiceen”`.
`on`	`“illa”`	`“lla”`
`onto`	`“ille”`	`“lle”`

Step #2: Since the Finnish language doesn’t use articles, delete all instances of the words “a” and “the”.

Input

You may assume that the input begins with a number, indicating how many messages to translate. Each subsequent line of input contains a message. A message may begin with a preposition. Messages will contain no punctuation and consist solely of lowercase letters [no punctuation]. All messages will be of length 80 or less, and will contain at least one word which is not an article. All prepositions will always be followed by two words.

Output

Your output should have one message per line. Be sure to capitalize the first word of the message, and include a period at the end of your message. When you output your messages, make sure you do not put any whitespace between the last word and the period.

Sample Input

```
10
put the turkey into the oven
now the turkey is in the oven
then take the turkey from the oven and put it onto the stove
now the turkey is ready to serve
yum yum yum put the turkey into your mouth
a fine kettle of finnish fish from the market
the pie
into the woods
grandmother lives on a farm over the river
the finnish chef hopes you have a happy thanksgiving
```

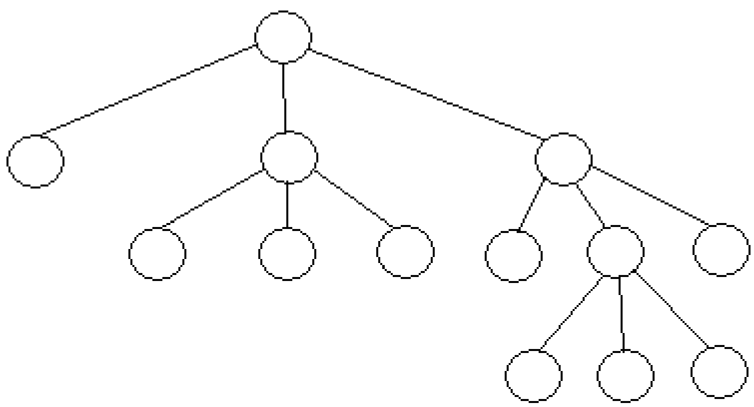
Output Corresponding to Sample Input

```
Put turkey ovenoon.
Now turkey is ovenissa.
Then take turkey ovenista and put it stovelle.
Now turkey is ready to serve.
Yum yum yum put turkey youroon mouthoon.
Fine kettle of finnish fish marketista.
Pie.
Woodsoon.
Grandmother lives farmilla over river.
Finnish chef hopes you have happy thanksgiving.
```

Problem 5
The Election

Triland is a little-known country that, just like the United States, is holding a presidential election this year. Like the U.S., Triland has a two-party system consisting of Democrats and Republicans. Similar to the U.S.'s electoral college system, under Triland's voting system it is possible for the candidate with the popular vote to lose the election. But Triland's voting system is very different from our electoral college system.

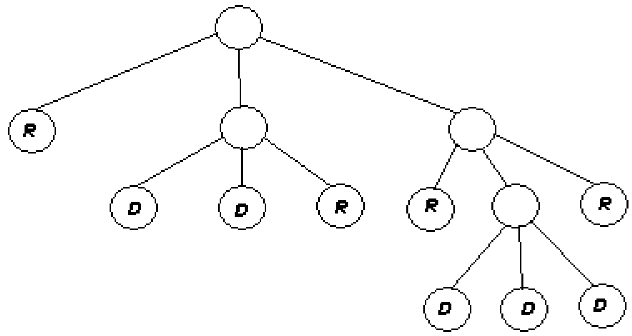
Triland is divided into three main blocs. Each bloc can either cast 1 vote (a voting bloc), or divide itself into three more sub-blocs (a subdivided bloc). Eventually all blocs bottom out into voting blocs. We can illustrate this with a tree structure, in which internal nodes are subdivided blocs and leaves are voting blocs. For example, in the tree shown below, the three main blocs of Triland are split into one voting bloc and two subdivided blocs, with the first subdivided bloc being divided into 3 voting blocs and the second subdivided bloc being further subdivided:



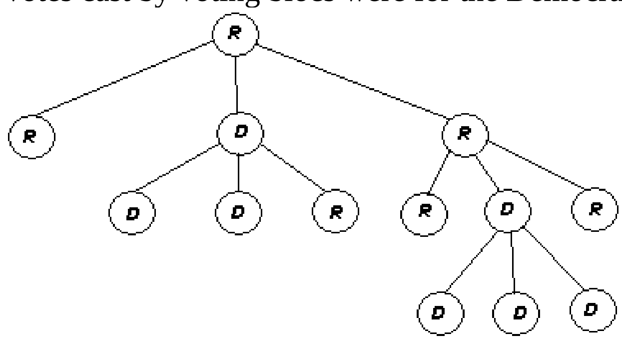
Triland is a dynamic place; its blocs are reconfigured often, which is why they need you to write a program to help them decide who wins an election.

When election time comes, the winner is decided as follows. Voting blocs cast their votes directly for either the Democrat (D) or the Republican (R). As soon as a subdivided bloc knows the outcome of its three sub-blocs, it propagates a vote as follows: if two (or more) of its immediate sub-blocs have voted Democrat, it propagates the Democrat vote; otherwise Republican. The vote propagated by the topmost subdivided bloc determines the winner.

Returning to the situation illustrated by the tree above, suppose the votes cast by the voting blocs are as follows (with D denoted Democrat and R denoting Republican):



Then the votes propagated up would be as follows, with the Republican candidate winning (even though a majority of the votes cast by voting blocs were for the Democrat):



Your task is to determine the winning party of the election, given the configurations of the voting blocs and the votes cast. Each configuration will be described using the following grammar:

$$C \rightarrow (BBB)$$
$$B \rightarrow D \mid R \mid (BBB)$$

For example, the configuration illustrated in the second figure above is described by:

(R(DDR)(R(DDD)R))

Input

The input will begin with an integer n on a line by itself, which denotes the number of test cases to follow. There will follow n lines, each containing the description of a case, as described above. There will be no blanks in the input.

Output

For each test case, print the case number and the winner, as illustrated below. Each case should be on a line by itself.

Sample Input

```
2
(R(DDR)(R(DDD)R))
(DRD)
```

Output Corresponding to Sample Input

```
Case 1: R
Case 2: D
```

Problem 6

Great Caesar Cipher

In cryptography, a Caesar cipher is one of the simplest and most widely known encryption techniques. It is a type of substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet. For example, with a shift of 3, A would be replaced by D, B would become E, and so on.

The method is named after Julius Caesar, who used it to communicate with his generals. The transformation can be represented by aligning two alphabets; the cipher alphabet is the plain alphabet rotated left or right by some number of positions. For instance, here is a Caesar cipher using a left rotation of three places (the shift parameter, here 3, is used as the key):

Plain: `ABCDEFGHIJKLMNOPQRSTUVWXYZ`

Cipher: `DEFGHIJKLMNOPQRSTUVWXYZABC`

When encrypting, a person looks up each letter of the message in the "plain" line and writes down the corresponding letter in the "cipher" line. Deciphering is done in reverse.

Input

The input is a sequence of test sets. Each set starts with a line containing a number *k* that is the shift parameter for the cipher, followed by a space, and the letter `E` for encrypt or letter `D` for decrypt. The second line in each set is a message of length 80 or less comprising uppercase and/or lowercase letters and spaces [no punctuation]. The last set starts with a *k* of value 0 indicating no more lines follow.

Output

For each message, output the encrypted or decrypted text message. The output should preserve the spaces and cases from the message.

Sample Input

```
3 E
The quick brown fox jumps over the lazy dog
3 D
Wkh txlfn eurzq ira mxpsv ryhu wkh odcb grj
2 D
Ocmg oa fca
0 E
```

Output Corresponding to Sample Input

```
Wkh txlfn eurzq ira mxpsv ryhu wkh odcb grj
The quick brown fox jumps over the lazy dog
Make my day
```

Problem 7

Stacks of Money

Mrs. Carol Adkins has designed a lesson for her elementary school class to teach them about money. At the moment, they are using fake \$5, \$10, and \$20 bills. Before the lesson starts, Mrs. Adkins arranges stacks of different bills for the students so that each one has the same amount of money. The total value of the bills in each stack does not exceed \$100. Sometimes, she takes one, and only one, bill from one student's stack, and puts it in the stack of another student. The students then have to work out which one has the most money, and which one has the least. Other times, Mrs. Atkins does not move a note so that each student has exactly the same amount of money. In this case, the students need to recognize and report this. Your task is to write a program to calculate the correct answer that Mrs. Adkins' students should report.

Input

The input consists of a series of scenarios for a number of lessons. The first line in each scenario consists of a positive integer, *N*, which represents the number of students in the class that day ($2 < N < 30000$). Each of the following *N* lines contains the data for one pupil. The lines contain four tokens, the first name of the student followed by the number of \$5, \$10 and \$20 notes (in that order) allocated to that student. Tokens are separated by single spaces. Input is terminated by a scenario where *N* equals *-1*. This scenario should not be processed. You may assume all students within one scenario have different names.

Output

Output consists of one line for each scenario. It will be in one of the following two formats where `X` and `Y` represent names.

```
 X has most, Y has least money.
All have the same amount.
```

Sample Input

```
5
Andy 0 0 2
Stephen 0 4 0
Eric 8 1 0
David 2 0 1
Phillip 0 2 1
3
Charles 0 3 0
Chip 0 1 1
Danny 4 1 0
-1
```

Output Corresponding to Sample Input

```
Eric has most, David has least money.
All have the same amount.
```