problem 3

Happy Numbers

4 points

JAVA program name must be prob03.java
C /C++ program name must be: prob03.exe

Introduction

Most humans think of numbers as cold and unfeeling. As a matter of fact, some numbers are happy, while others are unhappy. The procedure for learning whether a number is happy or unhappy is quite simple. Starting with any positive integer, replace the number with the sum of the squares of its digits. Repeat the process until the number equals one or loops endlessly in a cycle. Numbers that converge to one are happy numbers, while those that do not are unhappy numbers. For this program loop cycles will have less than a hundred entries.

For example, starting with the number 32, the sum of squares is 13 because 3x3 + 2x2 = 13. The sum of squares for 13 is 10 because 1x1 + 3x3 = 10. The sum of squares for 10 is 1, so 32 is a happy number.

Sample Input

The input will be a single positive integer. Several examples are given here.

```
32

4565

42

86

5555
```

Sample Output

The program must print a sentence indicating if the input number is happy or unhappy, then exit.

```
32 is a happy number

4565 is an unhappy number

42 is an unhappy number

86 is a happy number

5555 is a happy number
```

problem 5

Greek Numerals

5 points

JAVA program name must be prob05.java
C /C++ program name must be: prob05.exe

Introduction

The ancient Greeks used the letters of their alphabet to represent numbers. Decoding these numbers is an essential part of translating many ancient documents. The system shown in the chart on the right, known as Ionian numerals, used twenty-seven letters to represent the numbers 1-9, 10-90, and 100-900. Two-digit and three-digit numbers were recorded by writing the letters side by side. For example, the number twenty-seven is written KZ, a combination of the symbol for twenty, K, and the symbol for seven, Z.

Write a program to convert Greek numerals to decimal numbers.

Number Value	Greek Letter	Text Symbol
1	Α	A
2	Β	B
3	Γ	G
4	Δ	D
5	Ε	E
6	Ϝ	#
7	Ζ	Z
8	Η	Y
9	Θ	H
10	Ι	I
20	Κ	K
30	Λ	L
40	Μ	M
50	Ν	N
60	Ξ	X
70	Ο	O
80	Π	P
90	Ϙ	Q
100	Ρ	R
200	Σ	S
300	Τ	T
400	Υ	U
500	Φ	F
600	Χ	C
700	Ψ	$
800	Ω	W
900	Ϡ	3

Sample Input

Each line of input contains a sequence of text symbols that represent the Greek letters shown in the chart. The input ends with a single period character.

```
KZ
MB
3#
UQE
$L
.
```

Sample Output

The program must print the decimal number equivalent for each Greek numeral on a separate line.

```
27
42
906
495
730
```

problem 6
Product Review Site
6 points

JAVA program name must be prob06.java
C /C++ program name must be: prob06.exe

Introduction

A review web site allows users to rate products on a scale of one to five. These ratings are stored and displayed whenever other users search for the products. Ratings are displayed with a histogram showing how many of each rating the product has received and an average of all ratings. Write a program to display a ratings histogram and average from a set of ratings.

Sample Input

Each line of input is a single rating value from one to five. The input ends with the number zero.

```
4
5
3
4
2
4
3
5
2
1
3
2
4
3
2
4
3
3
2
4
3
3
5
3
1
4
3
2
4
5
4
3
4
3
3
4
3
0
```

Sample Output

The program must print a histogram that shows the number of occurrences of each rating value. The histogram must match the format shown below, where each line includes the rating value, the count in parentheses, the aligned vertical bar, and the number of '=' characters based on the count. The program must also print the average rating.

```
5 ( 4) |====
4 (11) |===========
3 (14) |==============
2 ( 6) |======
1 ( 2) |==
Average rating: 3.243243
```

Problem 3

Diamonds are Forever

3 points

JAVA: program name must be prob03.java
C /C++ program name must be: prob03.exe

Task Description

This mission, should you decide to accept it, involves drawing diamonds. Should be a simple coding assignment, no?

Impossible? No!

The inputs to your program are the character to use to draw the diamond, and the scale of the diamond. In this mission, scale is the height above and below the midline, including the midline.

Sample Input/Output

```
Character: a
Scale: 5

    a
   a a
  a   a
 a     a
a       a
 a     a
  a   a
   a a
    a
Done
```

Problem 4

Dirt Simple Calculator

4 points

JAVA: program name must be prob04.java
C /C++ program name must be: prob04.exe

Task Description

In years past, the CodeWars problem development team (TM) has designed a number of calculator programs that were, frankly, just way too complicated to be in the contest. This year, in an effort to make amends for the mistakes of the past, the team has provided this dirt simple calculator.

Write a program that functions as a simple calculator.

Input

Each line of the input is a series of integers separated by the operators '+', '-', or '*' and terminated by an equal sign '='. The program should continue to read input lines until it reads a line with only "0 =" on it.

Output

For each line of input, the program should print the result of the operations performed in order from left to right. The multiplication operation does not take precedence over addition and subtraction.

Sample Input

```
28 - 7 * 3 =
13 * 4 + 8 * 2 + 1 =
4 + 3 * 52 =
0 =
```

Sample Output

```
63
121
364
```

Problem 5

Going Green

4 points

JAVA: program name must be prob05.java
C /C++ program name must be: prob05.exe

Task Description

A research team has installed CO_2 monitoring equipment in a variety of locations around the globe. Data from these monitors is stored in a central database and the team produces reports on a monthly basis.

Write a program to find peak CO_2 levels in the measured data.

A peak is defined as a value that is greater than the value before it and the value after it..

Input

The input will contain multiple data sets. Each data set consists of two lines. The first line gives the month and year that the data was collected. The next line consists of a series of integers that represent the measured amount of CO_2 for a particular day. There will be one measurement reported per day. The input ends with a line that says "END 0".

Output

Your program must find peak CO_2 levels in the data. For each peak the program must print the date of the peak and the actual CO_2 measurement. If a peak spans two or more days, the program must print the entire date range of the peak.

Sample Input

```
JUNE 2003
48 52 47 46 44 42 43 44 47 49 44 43 39 40 40 36 31 36 36 37 42 51 59 62 62 67 75 70 66 69
FEBRUARY 2005
66 70 73 80 83 84 86 82 87 88 88 90 89 95 98 100 100 100 89 88 87 86 85 85 85 88 88 84
END 0
```

Sample Output

```
JUNE 2, 2003: 52
JUNE 10, 2003: 49
JUNE 14-15, 2003: 40
JUNE 27, 2003: 75
FEBRUARY 7, 2005: 86
FEBRUARY 12, 2005: 90
FEBRUARY 16-18, 2005: 100
FEBRUARY 26-27, 2005: 88
```

CODE WARS 2009

Problem 6

Kepler's Anomaly

5 points

JAVA: program name must be prob06.java
C /C++ program name must be: prob06.exe

Task Description

In the year 1605 Johannes Kepler discovered three basic laws of planetary motion. These laws are still used today for many applications where the positions of the planets need to be calculated. The first law states that a planet's orbit is not a circle but an ellipse with sun at one focus. The calculation of a planet's position require the use of variables with fancy names like "eccentricity" and "mean anomaly".

Luckily for you, for this program you won't need a degree in astrophysics. You don't even need to know what "eccentric anomaly" actually means. You will, however, have to write a program that can use iterations (in other words, a loop) to calculate a numeric value from a formula.

The eccentric anomaly E of a planet can be calculated from its eccentricity e and its mean anomaly M. Starting with an approximation E1 for E, a better approximation E2 can be calculated with this formula:

```
E2 = M + e * sin(E1)
```

The final value of E is computed by iterating (looping) the formula until the difference between E1 and E2 is less than a desired value v.

In Java and C++ the sin function expects an angle in radians, so all input and output angles will be in radians.

Sample Input

Each line of the input will contain the name of a planet, followed by four numbers: the values for M, e, E1, and v. The input ends with the word END and four zeros.

```
MERCURY 1.2 0.205635 1.391660 0.00001
VENUS 3.384683 0.0067547 4.18880 0.00001
PLUTO 4.170094 0.248808 4.660029 0.00001
END 0 0 0 0
```

Sample Output

For each planet the program must calculate the value of E until the absolute value of E2 - E1 is less than the value v. The program must then print the name of the planet and the final value for the eccentric anomaly.

```
MERCURY 1.40274
VENUS 3.38307
PLUTO 3.98436
```

Problem 8

Security Door

6 points

JAVA: program name must be prob08.java
C /C++ program name must be: prob08.exe

Task Description

Sid the intern has completed the initial wiring of a lab network in between sessions of eating snacks and texting his friends about cars and girls. The network connects via fiber optic cable to equipment in a wiring closet. Sid worries that one of the other interns could walk in and accidently disconnect the cable. He's seen this sort of thing before. Although Sid looks quite intimidating in his red robot boots and animé hero blue hair, he knows he can't guard the closet 24/7.

To keep the equipment secure, Sid the intern has devised a security system with an electronic door lock that can only be opened using a wireless remote. With the press of a button the remote transmits a code to a receiver inside the closet. The receiver verifies the code and opens the lock if it is valid. For added security the code changes each time the button is pressed according to this formula, known as a linear conguential generator:

```
          NewCode = ( 17 * OldCode + 91 ) modulus 256
```

Help Sid write a program for the receiver. After each valid code is received, the receiver must recognize as valid the next three codes in the sequence just in case Sid accidently presses the button while he's away from the door.

Input

The input will consist of a series of command-integer pairs. The SYNC command tells the receiver to use the following number as its "Old Code". The OPEN command requests the door to open using the following number as the security code. The input ends with the command END 0.

Output

When the program reads a SYNC command it must print the word SYNCHRONIZED. When the program reads an OPN command with a valid code it must print the word UNLOCK. If it reads an OPEN command with an invalid code it must print the word INVALID.

Sample Input

```
SYNC 37
OPEN 54
OPEN 241
OPEN 133
OPEN 66
SYNC 63
OPEN 133
END 0
```

Sample Output

```
SYNCHRONIZED
UNLOCK
UNLOCK
INVALID
UNLOCK
SYNCHRONIZED
UNLOCK
```

Task Description

We all know about palindromes. "Bob" is a simple palindrome. Did you know that you can generate numeric palindromes using a simple algorithm?

invent CODE WARS 2009

Problem 9

Fun With Palindromes

6 points

JAVA: program name must be prob09.java
C/C++ program name must be: prob09.exe

Algorithm to find numeric palindromes:

1. Input a number. Call it num1.
2. Set num2 equal to the reverse of num1.
3. Check to see if num1 equals num2. If so, output the number. Stop
4. Set num1 = num1 + num2.
5. Go to Step 2.

Here is an example:

```
181
   1. num1 = 181
   2. num2 = 181
   3. num1 equals num2. Stop
```

It took 0 complete traversals of the algorithm to find the numeric palindrome of 181. Total number of steps = 0

```
90
   1. num1 = 90
   2. num2 = 9
   3. num1 does not equal num2
   4. num1 = 90 + 9

Start at Step 2.
   2. num2 = 99
   3. num1 equals num2,  Stop
```

It took one full traversal of the algorithm to find the numeric palindrome of 90 (99). Total number of steps = 1.

Your task is to compute numeric palindromes for a given range of numbers.

Input

<Starting number> <Ending number>

Output

The number, number of steps used to calculate the palindrome, and the numeric palindrome.

Sample Input/Output

```
>> palindromes 929 936
Finding Palindromes for numbers from 929 to 936...
Number of Palindromes calculated: 8

|No.|Steps|Palindrome
|---|-----|---------------
|929|    0|929
|930|    1|969
|931|    2|1771
|932|    2|2882
|933|    2|3993
|934|    3|9119
|935|    4|25652
|936|    5|99099
```

problem 4

Reverse + Add = Palindrome

5 points

JAVA program name must be prob04.java
C/C++ program name must be: prob04.exe

Introduction

A simple way to generate numeric palindromes is the "reverse and add" method. Start with a positive integer, reverse the digits, and add it to the original number. If the result is not a palindrome (a sequence that is identical right-to-left and left-to-right), then repeat the process. For example, starting with 195 we add 591 (the reverse of the digits) to get 786, which is not a palindrome. Then we add 786 and 687 to get 1473, which is also not a palindrome. Then 1473 plus 3741 yields 5214, and 5214 plus 4125 yields 9339. The number 9339 is a palindrome, so we're done!

As a matter of fact, there are some rare numbers for which this process never yields a palindrome. For the purposes of this problem, you can assume that all input will yield a palindrome.

Sample Input

The input will be a single positive integer. Several examples are provided here.

```
195

304

628

570

259
```

Sample Output

The program must print the palindrome that results from applying the reverse and add method.

```
9339

707

5995

5115

2332
```

problem **8**
Letter Distribution
6 points

hp codewars 2012

Introduction

The use of letters in the English language is not evenly distributed. For example, the letters E and T are used far more often than the letters X and J In fact, the same principle holds true for any language in wide use. This same idea of inequitable distribution among a population can also been in in such disparate examples as the number of bids for items for sale on an auction web site, the distribution of wealth among people, and the scores of Code Wars participants. The study of distributions allows mathematicians to understand and work with dynamics and behaviors of large populations. It also helps computer system architects to design efficient systems for use by a large number of people.

We'll explore the idea of inequitable distributions by writing a program to display a histogram of letter occurrences sorted by popularity.

Sample Input

The input is a body of English text, up to 80 characters per line. The end of input is signaled by a single line with the string "###".

```
I have a dream that one day this nation will rise up and live out the true
meaning of its creed: "We hold these truths to be self-evident, that all men
are created equal." I have a dream that my four little children will one day
live in a nation where they will not be judged by the color of their skin but
by the content of their character.
###
```

Sample Output

The program must count the number of occurrences of each letter of the input and sort the letters by popularity, from most popular to least. Upper case and lower case letters are considered the same for counting purposes. Spaces and punctuation are to be ignored. Two or more letters with equal popularity must be sorted alphabetically. The program must print a horizontal histogram of the sorted letter counts as shown below so that one "*" is displayed for each occurrence of a letter.

```
E ***************************************
T *******************************
A **********************
I ********************
H ******************
L *****************
N *****************
R ****************
O ***************
D ************
U ********
C *******
S *******
Y ******
B *****
F *****
M *****
V *****
W *****
G **
J *
K *
P *
Q *
X
Z
```

problem **10**

Card Counting

7 points

Introduction

Casinos go through many decks of cards and are probably interested in periodically verifying that all expected 52 cards are there and no extras have managed to slip into the deck. You have decided to help them out with this problem, for a nominal fee.

Write a program that takes in a list of cards and then determines what, if anything is wrong with the deck. Missing cards (if any) should be listed first, followed by extra cards (if any) and the number of extras of that type. Output should be sorted by rank (2-A) and then suit (spades, hearts, diamonds, clubs) order.

Sample Input

The input is a set of lines of cards in the deck, not necessarily in order. Suits are abbreviated as follows: H=hearts, D=diamonds, C=clubs, S=spades. Thus, for example, "7S" is the seven of spades. The first line of input specifies the number of lines of card input to follow.

```
13
5D KC 3C 3D
10C 7D 4C AH
10S JH 6C 10D
7S 2H JS QH
2C 7C KD 8C
QD QS KH 2S
3H 4S 5H JD
3S 4H 5S 6H
8H 6S 7H QC
9C 8S 9H 6D
2D 8D AS 5C
JC AH 4D KS
AH 9S 10H 9D
```

Sample Output

```
Missing cards:
AD AC

Extra cards:
AH (2)
```

problem 8

Distinct Letters

4 points

Introduction

The word “uncopyrightable” has the interesting property that each of the fifteen letters it contains is only used once. In the English language, there is no word longer that uses distinct letters (each letter used in the word is used once, and once only).

There is one other 15 letter word with this property: “dermatoglyphics” (from ancient Greek derma = "skin", glyph = "carving"), which is the scientific study of fingerprints.

Write a program to determine if a word uses distinct letters.

Input

The input is a series of lines, with one word per line. The input ends with a single period.

```
UNCOPYRIGHTABLE
FLIPPER
EXECUTABLE
UNPROFITABLE
QUESTIONABLY
WINDOW
TAMBOURINE
.
```

Output

The program must print each word with the correct message; either "USES DISTINCT LETTERS" or "DOES NOT USE DISTINCT LETTERS.”

```
UNCOPYRIGHTABLE USES DISTINCT LETTERS
FLIPPER DOES NOT USE DISTINCT LETTERS
EXECUTABLE DOES NOT USE DISTINCT LETTERS
UNPROFITABLE USES DISTINCT LETTERS
QUESTIONABLY USES DISTINCT LETTERS
WINDOW DOES NOT USE DISTINCT LETTERS
TAMBOURINE USES DISTINCT LETTERS
```

problem 9

Counting Ones

5 points

hp codewars 2013

Introduction

This program is a fun math problem!

Consider a function f(n) that takes a positive integer n and returns the number of 1's in the decimal representation of all the integers from 0 to n, inclusive. For example, f(13) = 6, for the numbers 1, 10, 11 (twice, for two 1s), 12 and 13. Notice that f(1) = 1.

Your task is to write a program that calculates f(n).

Input

Each line of input is a single integer, up to a maximum of 9999. The input ends with a -1.

```
13
1
999
23
1111
9997
511
-1
```

Output

For each non-negative input, the program must print the value of f(n).

```
6
1
300
13
448
4000
204
```

problem 12 Morse Code Palindrome

6 points

Introduction

Let's combine two of our favorite memes: Palindromes and Morse code.

A	•–	J	•–––	S	•••
B	–•••	K	–•–	T	–
C	–•–•	L	•–••	U	••–
D	–••	M	––	V	•••–
E	•	N	–•	W	•––
F	••–•	O	–––	X	–••–
G	––•	P	•––•	Y	–•––
H	••••	Q	––•–	Z	––••
I	••	R	•–•		

Morse code

Remember that a palindrome is a string that is the same forward and backward, ignoring any spaces or punctuation. So the word "ATE", which is encoded as "•- - •", is a palindrome. So is the phrase "WE TAN".

Input

The input consists of one or more English words, written in upper case, per line. The input ends with a single period.

```
ELEGIZED
QUIRKILY
MERCURY
FACE A WINE
HAPPY DAY
FEVER REBEL
SOPRANOS
EMIT OLD UFO TIME
PROTEIN POWDER
ANNEXING
ENJOIN
.
```

Output

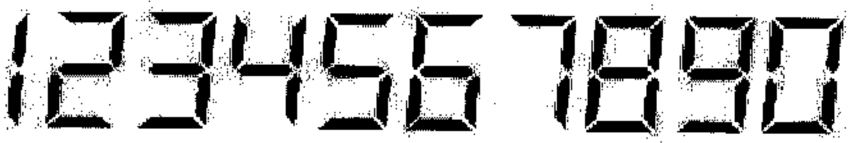
The program must convert the words to Morse code and determine if the pattern of dots and dashes is a palindrome. For each line of input, the program must print the line with the correct message, either "is a MCP", or "is *not* a MCP". Be sure to use the * characters to help the judges!

```
ELEGIZED is a MCP
QUIRKILY is a MCP
MERCURY is *not* a MCP
FACE A WINE is a MCP
HAPPY DAY is *not* a MCP
FEVER REBEL is a MCP
SOPRANOS is a MCP
EMIT OLD UFO TIME is a MCP
PROTEIN POWDER is *not* a MCP
ANNEXING is a MCP
ENJOIN is a MCP
```

problem 3 Clock Power

4 points

You amble over to the arena's scoreboard, where the event coordinators are trying to measure the power used by the clock. The clock uses four seven-segment LEDs to display the hours and the minutes. For example, as you can see in the image below, the number 1 requires lighting two segments, and the number 2 requires lighting five segments.



Lighting a single LED segment requires 15 milliamps. There is also a divider (:) between the hours and the minutes. Lighting the divider requires 20 milliamps. The first LED for the hours portion of the clock will not be lit for hour values less than 10.

Write a program to output the number of milliamps required to light the clock for a given a time value.

Input

Input begins with a single integer T (T < 50) representing the number of time values. The following T lines will contain a single time value of the format HH:MM. The HH value will be a number from 1 to 12. There will be no leading zero for hour values less than 10. The MM value will be a number from 0 to 59. The MM value will have a leading zero for values less than 10.

```
6
1:23
10:58
12:00
3:14
1:11
7:38
```

Output

For each time value, output the number of milliamps required.

```
200 milliamps
320 milliamps
305 milliamps
185 milliamps
110 milliamps
245 milliamps
```

problem 4
Candy Count
5 points

You stumble into the refreshment tent for some much needed nutrition, but you discover jars of candy instead.

On each table is one jar and a list of names and numbers. Apparently people have been making guesses about how much candy is in each jar. Now we need to know whose guesses were closest.

Write a program to determine the winner(s) with the closest guesses to the number of candies in each jar.

Input

Input for each jar begins with an integer C (C < 1000) describing the actual candy count for that jar.
The next line contains the number of people P (P < 30) guessing.

The following P lines contain the guess and name of the person. The guess is an integer G (G < 1000). A single space will separate the guess from the name. The name will consist only of uppercase and lowercase letters A-Z. Names will not contain any spaces.

Example 1:

```
480
4
90 John
400 Melinda
560 Chuck
173 Miika
```

Example 2:

```
362
5
123 Miika
456 John
321 Chuck
400 Melinda
314 David
```

Output

Output the name of the winner (the one with the closest guess for the jar.) If multiple people are just as close to the actual candy count, output their names in the order they appear in the input, separated by spaces.

Example 1:

```
Melinda Chuck
```

Example 2:

```
Melinda
```

problem 5

Pangram

5 points

You step into the next event tent, which has been decorated like a school classroom. Around the room are all the letters of the alphabet. On the large board at the front of the room, you see:

The quick brown fox jumps over the lazy dog.

The bulletin board on the wall explains: A sentence that uses every letter of the alphabet is called a pangram. A "perfect" pangram only uses each letter once.

Write a program to determine if a sentence is a pangram or a perfect pangram.

Input

Each line of input is a sentence that ends with a period. Other punctuation may also be included. The input ends with a single period.

```
THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG.
ALL YOUR BASE ARE BELONG TO US; SOMEONE SET US UP THE BOMB.
"NOW FAX A QUIZ TO JACK!" MY BRAVE GHOST PLED.
QUICK HIJINX SWIFTLY REVAMPED THE GAZEBO.
NEW JOB: FIX MR GLUCK'S HAZY TV, PDQ.
LOREM IPSUM DOLOR SIT AMET CONSECTETUR ADIPISCING ELIT.
PACK MY BOX WITH FIVE DOZEN LIQUOR JUGS.
.
```

Output

For each sentence, the program must print the key word PERFECT if the sentence is a perfect pangram, PANGRAM if it is a non-perfect pangram, or NEITHER if it is neither one. After the key word, the program must also print a colon, a space, and the original sentence.

```
PANGRAM: THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG.
NEITHER: ALL YOUR BASE ARE BELONG TO US; SOMEONE SET US UP THE BOMB.
PANGRAM: "NOW FAX A QUIZ TO JACK!" MY BRAVE GHOST PLED.
PANGRAM: QUICK HIJINX SWIFTLY REVAMPED THE GAZEBO.
PERFECT: NEW JOB: FIX MR GLUCK'S HAZY TV, PDQ.
NEITHER: LOREM IPSUM DOLOR SIT AMET CONSECTETUR ADIPISCING ELIT.
PANGRAM: PACK MY BOX WITH FIVE DOZEN LIQUOR JUGS.
```

problem 6
Don't Use No Double Negatives
5 points

Walking along the arena track, you almost don't see the misdirecting sign for the event that seems to want to not be discovered near the water table. Then you realize the misinformation is part of the problem! There is an instruction sheet that explains:

A double negative is a sentence which uses two negative words, such as "no", "not", or "don't". Double negatives are often used to emphasize the negative, but in other cases they cancel each other. Some people consider their usage bad grammar. Even when used correctly, we won't disagree that double negatives can make a sentence confusing. More extreme usage may include three, or four, or more negatives in a single statement. Multiple negatives can be used for comic effect or to obscure meaning, but don't say we didn't never warn you not to do it.

Write a program to count the number of negatives in a sentence. Use this list of negative words provided below. The sample input will demonstrate that there may or may not be other forms of negatives we won't be failing to ignore.

DON'T CAN'T ISN'T HAVEN'T CANNOT WOULDN'T COULDN'T
WON'T NO NOT NEVER NOBODY NOWHERE NEITHER AIN'T

Input

Each line of input is a single sentence that ends with a period. Sentences may have one or more of the following punctuation marks: , ; : ? ! "

The input ends with a single period.

```
THERE NEVER WAS NO MAN NOWHERE SO VIRTUOUS.
AXE YA EX WHAT TALKS IN TONGUES, SAY "NEVER MO NEITHER ME".
I HAVEN'T NEVER OWED NOTHING TO NO ONE.
BADGES? WE AIN'T GOT NO BADGES.
THIS IS A PERFECTLY POSITIVE SENTENCE.
I CAN'T GET NO, NO, NO, NO, HEY, HEY, HEY, NO SATISFACTION!
IT WOULDN'T BE INACCURATE TO ASSUME I COULDN'T SAY THAT I DON'T KNOW WHERE HE'S NOT.
.
```

Output

The program should print the number of negatives in each sentence, followed by a colon, space, and the sentence.

```
3: THERE NEVER WAS NO MAN NOWHERE SO VIRTUOUS.
2: AXE YA EX WHAT TALKS IN TONGUES, SAY "NEVER MO NEITHER ME".
3: I HAVEN'T NEVER OWED NOTHING TO NO ONE.
2: BADGES? WE AIN'T GOT NO BADGES.
0: THIS IS A PERFECTLY POSITIVE SENTENCE.
6: I CAN'T GET NO, NO, NO, NO, HEY, HEY, HEY, NO SATISFACTION!
4: IT WOULDN'T BE INACCURATE TO ASSUME I COULDN'T SAY THAT I DON'T KNOW WHERE HE'S NOT.
```

problem 8
N-Ultimate
6 points

You saunter into the next pavilion to see a seemingly endless series of words and numbers being displayed on the walls. You read this explanation on the central table:

The last element in a series may be called the *ultimate* element. The *penultimate* element is next-to last. So, by extension, the *N-ultimate* element is the Nth element from the end.

Write a program to find the N-ultimate element in a series.

Input

Each line of input starts with an integer N, followed by a series of N or more words/numbers/strings, terminated with a $ symbol. The input ends with the number zero and a $ symbol.

```
4 PROXIMATE DISTANT EXTREME FARTHEST ULTIMATE $
6 999 0 426 123 1337 31415 1414 5 321 $
2 WHO WHAT WHEN WHERE WHY HOW $
3 RED GREEN BLUE YELLOW ORANGE PURPLE BLACK WHITE $
7 GARCIA WANG ZHANG LI SMITH MULLER GONZALEZ SMIRNOV NGUYEN HERNANDEZ $
0 $
```

Output

For each line of input the program must print the N-ultimate word.

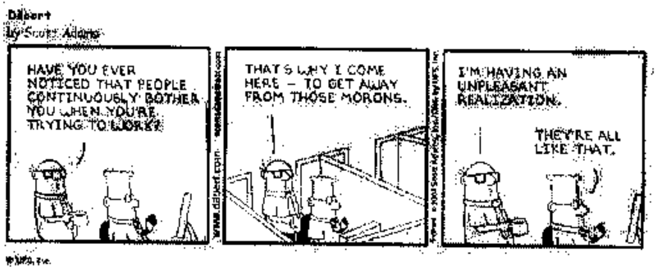
```
DISTANT
123
WHY
PURPLE
LI
```

Problem 8 Counting Cubes [5 points]

Summary

We've just opened up a new facility that has 300 new spaces available for our employees. We're now in the process of identifying the employees that are moving into this new space. We have a cube assignment list and want to ensure that we're using the space effectively. In order to know that, we need to find out if there are any empty cubes, any duplicate cube assignments, and any unassigned cubes. Write a program that can analyze the current list and provide answers to those questions.

The first line of input will contain the number of employee to cube mappings that follow. Each employee to cube mapping will consist of the employee's first name followed by a single space and a cube location (0 to 300). An employee with cube location of 0 does not have an assigned cube. If a cube is empty the employee name will be "NA". The employee names are unique and we have ensured that each employee is listed only once.


Dilbert
by Scott Adams
HAVE YOU EVER NOTICED THAT PEOPLE CONTINUOUSLY BOTHER YOU WHEN YOU'RE TRYING TO WORK?
THAT'S WHY I COME HERE – TO GET AWAY FROM THOSE MORONS.
I'M HAVING AN UNPLEASANT REALIZATION.
THEY'RE ALL LIKE THAT.
www.dilbert.com
© UFS, Inc.

The output will consist of the number of empty cubes, the number of duplicate cube assignments (two or more employees assigned to the same cube), and the number of unassigned cubes.

Sample Input

```
16
Joe 11
Bob 123
NA 101
Katy 125
Sam 47
Mike 59
NA 23
Vivek 62
Fred 0
Lars 74
Oscar 86
Caroline 11
NA 90
Erin 11
Rachel 111
Nate 125
```

Sample Output

```
Empty Cubes: 3
Duplicate Cube Assignments: 2
Employees without Cube: 1
```

Problem 3 Queen Ann [3 points]

Summary

Queen Ann likes kittens, but she hates cats. She likes puppies, but she hates dogs. Queen Ann likes spoons, but not forks or knives. She likes summer, but she doesn't like heat or sunshine. She hates winter, but she likes freezing blizzards. Queen Ann likes pepper but not salt, and pizza but not pasta.

Without looking ahead, can you solve the riddle of what kinds of things Queen Ann likes or does not like?

** SPOILER ALERT **

Here is the solution to the riddle: Queen Ann only likes words that have double-letters, like her name. Write a program that can tell if a word is something that Queen Ann likes.

Input

The first line of input specifies how many words the program must read. Each word follows on a separate line. Words are composed of upper case English letters.

```
7
KITTENS
FORKS
WINTER
RIDDLES
TELEVISION
BOOKS
COWS
```

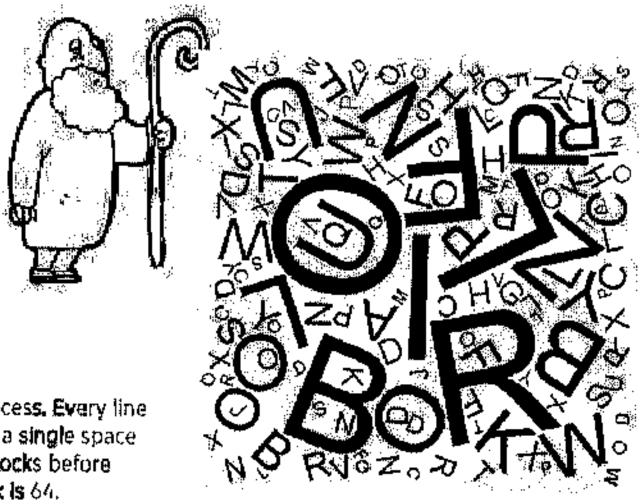
Output

For each input word, the program must print whether Queen Ann "likes" or "hates" the word.

```
likes KITTENS
hates FORKS
hates WINTER
likes RIDDLES
hates TELEVISION
likes BOOKS
hates COWS
```

Problem 5 Letter Tax [4 points]

Summary



Every year, local letter herders bring their flocks into town to pay their letter tax. The tax rate is variable based on other taxes paid by the herders. Don't get them started complaining about the cryptographic tax code or the irrational number tax -- there's no end to it.

Here's how the tax works: the tax officer gives the letter herder an integer number N. The letter herder must relinquish every Nth letter to the tax, starting with the first letter and separating out every Nth letter afterward.
Write a program to print the state of the flocks after the tax.

Input

The first line of input indicates the number of flocks the program will process. Every line after that begins with an integer (the tax rate N) and then is followed by a single space and a sequence of English letters. These are the representations of the flocks before the tax. As you probably already know, the maximum size of a letter flock is 64.

```
3
4 xCORxRECxT
5 agoodEbyeLDetteKrs
11 xTheQuickBrwownFoxJumpesOverTheLaxzyDog
```

Output

For each flock, the program must print the representation of the flock after the tax. The letters must be printed in the same order as the input, but without the taxed letters. The program must also print (on the same line) the number of letters remaining in the flock after the tax. The letters removed must start with the first letter in each flock and every Nth letter must be removed afterward.

```
CORRECT 7
goodbyeLetters 14
TheQuickBrownFoxJumpsOverTheLazyDog 35
```

Problem 7 Share Your Letters [5 points]

Summary

You may have heard that you learned everything you needed to know in kindergarten. If that's true, then why are you still in school? Does that seem right to you?

At CodeWars, we believe that you actually do learn quite a few useful things in school after kindergarten. For example, you probably learned to write programs sometime after kindergarten. However, one of the useful lessons we all should have learned, but we sometimes forget, is that it's nice to share your things with others. With that in mind, your task for this problem is to write a program that can determine the set of letters shared in common among three words.

Input

The first line of input indicates the number of word triplets the program must read. Each line after contains three words, and we use that term loosely, separated by one or more spaces. Don't assume there will only be one space between words.

```
3
TEST         MEANT           TIME
KINDERGARTEN CHICKENFEATHERS SPECIALITIES
ABSURD       SUBORDINATE     DUMBELLS3
```

Output

For each triplet, the program must print all the letters that are shared by all three words. If a letter appears multiple times in each word, then it should also appear the corresponding number of times in the output. The letters must be printed in alphabetic order.

```
ET
AEEIT
BDSU
```