23. 369 - Combinations

Time limit: 3.000 seconds

Combinations

Computing the exact number of ways that N things can be taken M at a time can be a great challenge when N and/or M become very large. Challenges are the stuff of contests. Therefore, you are to make just such a computation given the following:

GIVEN:

$$5 \le N \le 100, \quad \text{and} \quad 5 \le M \le 100, \quad \text{and} \quad M \le N$$

Compute the **EXACT** value of:

$$C = \frac{N!}{(N-M)! \times M!}$$

You may assume that the final value of C will fit in a 32-bit Pascal LongInt or a C long.

For the record, the exact value of 100! is:
93,326,215,443,944,152,681,699,238,856,266,700,490,715,968,264,381,621,
468,592,963,895,217,599,993,229,915,608,941,463,976,156,518,286,253,
697,920,827,223,758,251,185,210,916,864,000,000,000,000,000,000,000,000

Input and Output

The input to this program will be one or more lines each containing zero or more leading spaces, a value for N, one or more spaces, and a value for M. The last line of the input file will contain a dummy N, M pair with both values equal to zero. Your program should terminate when this line is read.

The output from this program should be in the form:

N `things taken` M `at a time is` C `exactly.`

Sample Input

```
    100   6
     20   5
     18   6
      0   0
```

Sample Output

```
100 things taken 6 at a time is 1192052400 exactly.
20 things taken 5 at a time is 15504 exactly.
18 things taken 6 at a time is 18564 exactly.
```

32. 10327 - Flip Sort

Time limit: 3.000 second

Flip Sort

Sorting in computer science is an important part. Almost every problem can be solved effeciently if sorted data are found. There are some excellent sorting algorithm which has already acheived the lower bound nlgn. In this problem we will also discuss about a new sorting approach. In this approach only one operation (Flip) is available and that is you can exchange two adjacent terms. If you think a while, you will see that it is always possible to sort a set of numbers in this way.

The Problem

A set of integers will be given. Now using the above approach we want to sort the numbers in ascending order. You have to find out the minimum number of flips required. Such as to sort "1 2 3" we need no flip operation whether to sort "2 3 1" we need at least 2 flip operations.

The Input

The input will start with a positive integer N (N<=1000). In next few lines there will be N integers. Input will be terminated by EOF.

The Output

For each data set print "Minimum exchange operations : M" where M is the minimum flip operations required to perform sorting. Use a seperate line for each case.

Sample Input

```
3
1 2 3
3
2 3 1
```

Sample Output

```
Minimum exchange operations : 0
Minimum exchange operations : 2
```

Md. Kamruzzaman

34. 10346 - Peter's Smokes

Time limit: 3.000 seconds

Problem B
Peter's Smokes
Input: standard input
Output: standard output
Time Limit: 4 seconds
Memory Limit: 32 MB

Peter has **n** cigarettes. He smokes them one by one keeping all the butts. Out of **k > 1**butts he can roll a new cigarette.

How many cigarettes can Peter have?

Input

Input is a sequence of lines. Each line contains two integer numbers giving the values of**n** and **k**. The input is terminated by end of file.

Output

For each line of input, output one integer number on a separate line giving the maximum number of cigarettes that Peter can have.

Sample Input

```
4 3
10 3
100 5
```

Sample Output

```
5
14
124
```

Source: University of Alberta Local Contest

36. 10370 - Above Average

Time limit: 3.000 seconds

Problem D: Above Average

It is said that 90% of frosh expect to be above average in their class. You are to provide a reality check.

The first line of standard input contains an integer C, the number of test cases. C data sets follow. Each data set begins with an integer, N, the number of people in the class (1 <= N <= 1000). N integers follow, separated by spaces or newlines, each giving the final grade (an integer between 0 and 100) of a student in the class. For each case you are to output a line giving the percentage of students whose grade is above average, rounded to 3 decimal places.

Sample Input

```
5
5 50 50 70 80 100
7 100 95 90 80 70 60 50
3 70 90 80
3 70 90 81
9 100 99 98 97 96 95 94 93 91
```

Output for Sample Input

```
40.000%
57.143%
33.333%
66.667%
55.556%
```

39. 10420 - List of Conquests

Time limit: 3.000 seconds

Problem B
List of Conquests
Input: standard input
Output: standard output
Time Limit: 2 seconds

In Act I, Leporello is telling Donna Elvira about his master's long list of conquests:

> ``This is the list of the beauties my master has loved, a list I've made out myself: take a look, read it with me. In Italy six hundred and forty, in Germany two hundred and thirty-one, a hundred in France, ninety-one in Turkey; but in Spain already a thousand and three! Among them are country girls, waiting-maids, city beauties; there are countesses, baronesses, marchionesses, princesses: women of every rank, of every size, of every age.'' (*Madamina, il catalogo è questo*)

As Leporello records all the ``beauties'' Don Giovanni ``loved'' in chronological order, it is very troublesome for him to present his master's conquest to others because he needs to count the number of ``beauties'' by their nationality each time. You are to help Leporello to count.

Input

The input consists of at most **2000** lines, but the first. The first line contains a number **n**, indicating that there will be **n** more lines. Each following line, with at most **75**characters, contains a country (the first word) and the name of a woman (the rest of the words in the line) Giovanni loved. You may assume that the name of all countries consist of only one word.

Output

The output consists of lines in alphabetical order. Each line starts with the name of a country, followed by the total number of women Giovanni loved in that country, separated by a space.

Sample Input

```
3
Spain Donna Elvira
England Jane Doe
Spain Donna Anna
```

Sample Output

```
England 1
Spain 2
```

51. 10591 - Happy Number

Time limit: 3.000 seconds

Problem C	**Happy Number**
Time Limit	1 Second

Let the sum of the square of the digits of a positive integer $\mathbf{S_0}$ be represented by $\mathbf{S_1}$. In a similar way, let the sum of the squares of the digits of $\mathbf{S_1}$ be represented by $\mathbf{S_2}$ and so on. If $\mathbf{S_i} = 1$ for some $\mathbf{i} \geq 1$, then the original integer $\mathbf{S_0}$ is said to be Happy number. A number, which is not happy, is called Unhappy number. For example 7 is a Happy number since 7 -> 49 -> 97 -> 130 -> 10 -> 1 and 4 is an Unhappy number since 4 -> 16 -> 37 -> 58 -> 89 -> 145 -> 42 -> 20 -> 4.

Input

The input consists of several test cases, the number of which you are given in the first line of the input. Each test case consists of one line containing a single positive integer**N** smaller than $\mathbf{10^9}$.

Output

For each test case, you must print one of the following messages:

```
Case #p: N is a Happy number.
Case #p: N is an Unhappy number.
```

Here **p** stands for the case number (starting from 1). You should print the first message if the number **N** is a happy number. Otherwise, print the second line.

Sample Input	Output for Sample Input
3 7 4 13	Case #1: 7 is a Happy number. Case #2: 4 is an Unhappy number. Case #3: 13 is a Happy number.

Problemsetter: Mohammed Shamsul Alam
International Islamic University Chittagong
Special thanks to Muhammad Abul Hasan

62. 10789 - Prime Frequency

Time limit: 3.000 seconds

Prime Frequency

Given a string containing only alpha-numerals (0-9, A-Z and a-z) you have to count the frequency (the number of times the character is present) of all the characters and report only those characters whose frequency is a prime number. A prime number is a number, which is divisible by exactly two different integers. Some examples of prime numbers are **2**, **3**, **5**, **7**, **11** etc.

Input

The first line of the input is an integer ***T*** (0 < ***T*** < 201) that indicates how many sets of inputs are there. Each of the next ***T*** lines contains a single set of input.

The input of each test set is a string consisting alpha-numerals only. The length of this string is positive and less than **2001**.

Output

For each set of input produce one line of output. This line contains the serial of output followed by the characters whose frequency in the input string is a prime number. These characters are to be sorted in lexicographically ascending order. Here ``lexicographically ascending" means ascending in terms of the **ASCII** values. Look at the output for sample input for details. If none of the character frequency is a prime number, you should print `empty' (without the quotes) instead.

Sample Input

```
3
ABCC
AABBBBDDDDD
ABCDFFFF
```

Sample Output

```
Case 1: C
Case 2: AD
Case 3: empty
```

64. 10812 - Beat the Spread!

Time limit: 3.000 seconds

Problem D: Beat the Spread!

Superbowl Sunday is nearly here. In order to pass the time waiting for the half-time commercials and wardrobe malfunctions, the local hackers have organized a betting pool on the game. Members place their bets on the sum of the two final scores, or on the absolute difference between the two scores.

Given the winning numbers for each type of bet, can you deduce the final scores?

The first line of input contains *n*, the number of test cases. *n* lines follow, each representing a test case. Each test case gives *s* and *d*, non-negative integers representing the sum and (absolute) difference between the two final scores. For each test case, output a line giving the two final scores, largest first. If there are no such scores, output a line containing "impossible". Recall that football scores are always non-negative integers.

Sample Input

```
2
40 20
20 40
```

Output for Sample Input

```
30 10
impossible
```

Gordon V. Cormack

66. 10921 - Find the Telephone

Time limit: 3.000 seconds

Time Limit: 1 second

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 $1 \le C \le 30$. 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
```

68. 10924 - Prime Words

Time limit: 3.000 seconds

Problem E - Prime Words

Time Limit: 1 second

A prime number is a number that has only two divisors: itself and the number one. Examples of prime numbers are: 1, 2, 3, 5, 17, 101 and 10007.

In this problem you should read a set of words, each word is composed only by letters in the range **a-z** and **A-Z**. Each letter has a specific value, the letter **a** is worth **1**, letter**b** is worth **2** and so on until letter **z** that is worth **26**. In the same way, letter **A** is worth**27**, letter **B** is worth **28** and letter **Z** is worth **52**.

You should write a program to determine if a word is a prime word or not. A word is a prime word if the sum of its letters is a prime number.

Input

The input consists of a set of words. Each word is in a line by itself and has **L** letters, where **1 ≤ L ≤ 20**. The input is terminated by enf of file (EOF).

Output

For each word you should print: **It is a prime word.**, if the sum of the letters of the word is a prime number, otherwise you should print: **It is not a prime word.**.

Sample Input

```
UFRN
contest
AcM
```

Sample Output

```
It is a prime word.
It is not a prime word.
It is not a prime word.
```

Problem setter: S�rgio Queiroz de Medeiros

33. 445 - Marvelous Mazes

Time limit: 3.000 seconds

Marvelous Mazes

Your mission, if you decide to accept it, is to create a maze drawing program. A maze will consist of the alphabetic characters `A-Z`, `*` (asterisk), and spaces.

Input and Output

Your program will get the information for the mazes from the input file. This file will contain lines of characters which your program must interpret to draw a maze. Each row of the maze will be described by a series of numbers and characters, where the numbers before a character tell how many times that character will be used. If there are multiple digits in a number before a character, then the number of times to repeat the character is the sum of the digits before that character.

The lowercase letter "`b`" will be used in the input file to represent spaces in the maze. The descriptions for different rows in the maze will be separated by an exclamation point (`!`) or by an end of line.

Descriptions for different mazes will be separated by a blank line in both input and output. The input file will be terminated by an end of file.

There is no limit to the number of rows in a maze or the number of mazes in a file, though no row will contain more than 132 characters.

Happy mazing!

Sample Input

```
1T1b5T!1T2b1T1b2T!1T1b1T2b2T!1T3b1T1b1T!3T3b1T!1T3b1T1b1T!5T1*1T

11X21b1X
4X1b1X
```

Sample Output

```
T TTTTT
T  T TT
T T  TT
T   T T
TTT   T
T   T T
TTTTT*T

XX   X
XXXX X
```

70. 10931 - Parity

Time limit: 3.000 seconds

Problem E - Parity

Time Limit: 1 second

We define the parity of an integer **n** as the sum of the bits in binary representation computed modulo two. As an example, the number *21 = 10101*$_2$ has three *1s* in its binary representation so it has parity *3 (mod 2)*, or *1*.

In this problem you have to calculate the parity of an integer **1 ≤ I ≤ 2147483647**.

Input

Each line of the input has an integer **I** and the end of the input is indicated by a line where **I = 0** that should not be processed.

Output

For each integer **I** in the inputt you should print a line **The parity of B is P (mod 2).**, where **B** is the binary representation of **I**.

Sample Input

```
1
2
10
21
0
```

Sample Output

```
The parity of 1 is 1 (mod 2).
The parity of 10 is 1 (mod 2).
The parity of 1010 is 2 (mod 2).
The parity of 10101 is 3 (mod 2).
```

Problem setter: S�rgio Queiroz de Medeiros

5. 11172 - Relational Operator

Time limit: 3.000 seconds

Some operators checks about the relationship between two values and these operators are called relational operators. Given two numerical values your job is just to find out the relationship between them that is (i) First one is greater than the second (ii) First one is less than the second or (iii) First and second one is equal.

Input

First line of the input file is an integer t (t<15) which denotes how many sets of inputs are there. Each of the next t lines contain two integers a and b (|a|,|b|<1000000001).

Output

For each line of input produce one line of output. This line contains any one of the relational operators ">", "<" or "=", which indicates the relation that is appropriate for the given two numbers.

Sample Input

```
3
10 20
20 10
10 10
```

Output for Sample Input

```
<

>

=
```

6. 11185 - Ternary

Time limit: 3.000 seconds

You will be given a decimal number. You will have to convert it to its ternary (Base 3) equivalent.

Input

The input file contains at most 100 lines of inputs. Each line contains a non-negative decimal integer N(N<1000000001). Input is terminated by a line containing a negative value. This line should not be processed.

Output

For each line of input produce one line of output. This line contains the ternary equivalent of decimal value N.

Sample Input

```
10
100
1000
-1
```

Output for Sample Input

```
101
10201
1101001
```

5. 11172 - Relational Operator

Time limit: 3.000 seconds

Some operators checks about the relationship between two values and these operators are called relational operators. Given two numerical values your job is just to find out the relationship between them that is (i) First one is greater than the second (ii) First one is less than the second or (iii) First and second one is equal.

Input

First line of the input file is an integer t (t<15) which denotes how many sets of inputs are there. Each of the next t lines contain two integers a and b (|a|,|b|<1000000001).

Output

For each line of input produce one line of output. This line contains any one of the relational operators “>”, “<” or “=”, which indicates the relation that is appropriate for the given two numbers.

Sample Input

```
3
10 20
20 10
10 10
```

Output for Sample Input

```
<

>

=
```

6. 11185 - Ternary

Time limit: 3.000 seconds

You will be given a decimal number. You will have to convert it to its ternary (Base 3) equivalent.

Input

The input file contains at most 100 lines of inputs. Each line contains a non-negative decimal integer N(N<1000000001). Input is terminated by a line containing a negative value. This line should not be processed.

Output

For each line of input produce one line of output. This line contains the ternary equivalent of decimal value N.

Sample Input

```
10
100
1000
-1
```

Output for Sample Input

```
101
10201
1101001
```

7. 11192 - Group Reverse

Time limit: 3.000 seconds

Problem G
Group Reverse
Input: Standard Input

Output: Standard Output

Group reversing a string means reversing a string by groups. For example consider a string:

“**TOBENUMBERONEWEMEETAGAINANDAGAINUNDERBLUEICPCSKY**”

This string has length 48. We have divided into 8 groups of equal length and so the length of each group is 6. Now we can reverse each of these eight groups to get a new string:

“UNEBOTNOREBMEEMEWENIAGATAGADNAEDNUNIIEULBRYKSCPC”

Given the string and number of groups in it, your program will have to group reverse it.

Input

The input file contains at most 101 lines of inputs. Each line contains at integer G (G<10) which denotes the number of groups followed by a string whose length is a multiple of G. The length of the string is not greater than 100. The string contains only alpha numerals. Input is terminated by a line containing a single zero.

Output

For each line of input produce one line of output which contains the group reversed string.

Sample Input	Output for Sample Input
3 ABCEHSHSH	CBASHEHSH
5 FA0ETASINAHGRI0NATWON0QA0NARI0	ATE0AFGHANISTAN0IRAQ0NOW0IRAN0
0	

Problem-setter: Shahriar Manzoor

Special Thanks: Derek Kisman

12. 11332 - Summing Digits

Time limit: 3.000 seconds

Problem J: Summing Digits

For a positive integer n, let f(n) denote the sum of the digits of nwhen represented in base 10. It is easy to see that the sequence of numbers n, f(n), f(f(n)), f(f(f(n))), ... eventually becomes a single digit number that repeats forever. Let this single digit be denoted g(n).

For example, consider n = 1234567892. Then:

```
f(n) = 1+2+3+4+5+6+7+8+9+2 = 47
f(f(n)) = 4+7 = 11
f(f(f(n))) = 1+1 = 2
```

Therefore, g(1234567892) = 2.

Each line of input contains a single positive integer n at most 2,000,000,000. For each such integer, you are to output a single line containing g(n). Input is terminated by n = 0 which should not be processed.

Sample input

```
2
11
47
1234567892
0
```

Output for sample input

```
2
2
2
2
```

Zachary Friggstad

15. 11388 - GCD LCM

Time limit: 1.000 seconds

Problem D: GCD LCM

Input: standard input
Output: standard output

The GCD of two positive integers is the largest integer that divides both the integers without any remainder. The LCM of two positive integers is the smallest positive integer that is divisible by both the integers. A positive integer can be the GCD of many pairs of numbers. Similarly, it can be the LCM of many pairs of numbers. In this problem, you will be given two positive integers. You have to output a pair of numbers whose GCD is the first number and LCM is the second number.

Input

The first line of input will consist of a positive integer **T**. **T** denotes the number of cases. Each of the next **T** lines will contain two positive integer, **G** and **L.**

Output

For each case of input, there will be one line of output. It will contain two positive integers **a** and **b**, **a ≤ b**, which has a GCD of **G** and LCM of **L.** In case there is more than one pair satisfying the condition, output the pair for which **a** is minimized. In case there is no such pair, output -1.

Constraints

- **T ≤ 100**
- Both **G** and **L** will be less than $\mathbf{2^{31}}$.

Sample Input	Output for Sample Input

2
1 2
3 4
	1 2
	-1

Problem setter: Shamim Hafiz

16. 11417 - GCD

Time limit: 2.000 seconds

Given the value of N, you will have to find the value of G. The definition of G is given below:

$$G = \sum_{i=1}^{i<N} \sum_{j=i+1}^{j \le N} GCD(i, j)$$

Here GCD(i,j) means the greatest common divisor of integer i and integer j.

For those who have trouble understanding summation notation, the meaning of G is given in the following code:

```
G=0;

for(i=1;i<N;i++)

for(j=i+1;j<=N;j++)

{

    G+=GCD(i,j);

}

/*Here  GCD()  is  a  function  that  finds  the  greatest
common divisor of the two input numbers*/
```

Input

The input file contains at most 100 lines of inputs. Each line contains an integer N (1<N<501). The meaning of N is given in the problem statement. Input is terminated by a line containing a single zero. This zero should not be processed.

Output

For each line of input produce one line of output. This line contains the value of G for corresponding N.

Sample Input	Output for Sample Input
10	67
100	13015
500	442011
0	

18. 11462 - Age Sort

Time limit: don’t know

You are given the ages (in years) of all people of a country with at least 1 year of age. You know that no individual in that country lives for 100 or more years. Now, you are given a very simple task of sorting all the ages in ascending order.

Input

There are multiple test cases in the input file. Each case starts with an integer **n** (0<**n**<=2000000), the total number of people. In the next line, there are **n** integers indicating the ages. Input is terminated with a case where **n** = 0. This case should not be processed.

Output

For each case, print a line with **n** space separated integers. These integers are the ages of that country sorted in ascending order.

Warning: Input Data is pretty big (~ 25 MB) so use faster IO.

Sample Input	Output for Sample Input

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

Note: The memory limit of this problem is 2 Megabyte Only.

Problem Setter: Mohammad Mahmudur Rahman

Special Thanks: Shahriar Manzoor

20. 11530 - SMS Typing

Time limit: 1.000 seconds

Cell phones have become an essential part of modern life. In addition to making voice calls, cell phones can be used to send text messages, which are known as SMS for short. Unlike computer keyboards, most cell phones have limited number of keys. To accommodate all alphabets, letters are compacted into single key. Therefore, to type certain characters, a key must be repeatedly pressed until that character is shown on the display panel.

In this problem we are interested in finding out the number of times keys on a cell phone must be pressed to type a particular message.

In this problem we will assume that the key pad of our cell phone is arranged as follows.

```
---------------------

|      | abc | def  |

---------------------

| ghi  | jkl | mno  |

---------------------

| pqrs | tuv | wxyz |

---------------------

|      | <SP>|      |

---------------------
```

In the above grid each cell represents one key. Here **SP** means a space. In order to type the letter 'a', we must press that key once, however to type 'b' the same key must be repeatedly pressed twice and for 'c' three times. In the same manner, one key press for 'd', two for 'e' and three for 'f'. This is also applicable for the remaining keys and letters. Note that it takes a single press to type a space.

Input

The first line of input will be a positive integer **T** where **T** denotes the number of test cases. **T** lines will then follow each containing only spaces and lower case letters. Each line will contain at least 1 and at most 100 characters.

Output

For every case of input there will be one line of output. It will first contain the case number followed by the number of key presses required to type the message of that case. Look at the sample output for exact formatting.

Sample Input	**Output for Sample Input**
2	Case #1: 29
welcome to ulab	Case #2: 41
good luck and have fun	

2. 10008 - What's Cryptanalysis?

Time limit: 3.000 seconds

What's Cryptanalysis?

Cryptanalysis is the process of breaking someone else's cryptographic writing. This sometimes involves some kind of statistical analysis of a passage of (encrypted) text. Your task is to write a program which performs a simple analysis of a given text.

Input

The first line of input contains a single positive decimal integer *n*. This is the number of lines which follow in the input. The next *n* lines will contain zero or more characters (possibly including whitespace). This is the text which must be analyzed.

Output

Each line of output contains a single uppercase letter, followed by a single space, then followed by a positive decimal integer. The integer indicates how many times the corresponding letter appears in the input text. Upper and lower case letters in the input are to be considered the same. No other characters must be counted. The output must be sorted in descending count order; that is, the most frequent letter is on the first output line, and the last line of output indicates the least frequent letter. If two letters have the same frequency, then the letter which comes first in the alphabet must appear first in the output. If a letter does not appear in the text, then that letter must not appear in the output.

Sample Input

```
3
This is a test.
Count me 1 2 3 4 5.
Wow!!!!  Is this question easy?
```

Sample Output

```
S 7
T 6
I 5
E 4
O 3
A 2
H 2
N 2
U 2
W 2
C 1
M 1
Q 1
Y 1
```

26. 11827 - Maximum GCD

Time limit: 1.000 seconds

Given the *N* integers, you have to find the maximum GCD(greatest common divisor) of every possible pair of these integers.

Input

The first line of input is an integer *N*(1<*N*<*100*) that determines the number of test cases.

The following *N* lines are the *N* test cases. Each test case contains *M* (1<*M*<*100*)positive integers that you have to find the maximum of GCD.

Output

For each test case show the maximum GCD of every possible pair.

Sample Input	Output for Sample Input

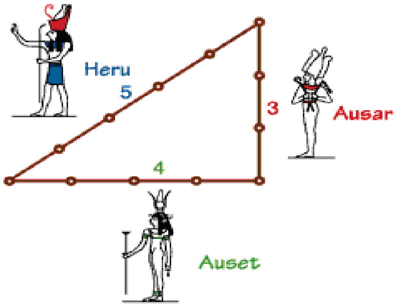
3
10 20 30 40
7 5 12
125 15 25
	20
	1
	25

27. 11854 - Egypt

Time limit: 1.000 seconds

Problem A: Egypt

A long time ago, the Egyptians figured out that a triangle with sides of length 3, 4, and 5 had a right angle as its largest angle. You must determine if other triangles have a similar property.



The Input

Input represents several test cases, followed by a line containing 0 0 0. Each test case has three positive integers, less than 30,000, denoting the lengths of the sides of a triangle.

The Output

For each test case, a line containing "right" if the triangle is a right triangle, and a line containing "wrong" if the triangle is not a right triangle.

Sample Input

```
6 8 10
25 52 60
5 12 13
0 0 0
```

Output for Sample Input

```
right
wrong
right
```

28. 11900 - Boiled Eggs

Time limit: 1.000 seconds

Three of the trouble-makers went to Malaysia this year. A rest house was booked for them. Unlike other rest houses, this rest house was like a normal duplex house. So, it had a kitchen. And the trouble-makers were given all the ingredients to cook, but they had to cook themselves.

None of them had any previous cooking experience, but they became very excited and planned to cook so many delicious foods! Ideas were coming from their minds like rains from clouds. So, they went to the super market and bought a lot of extra ingredients for their great recipes. For example, they bought **20** eggs. The excited trouble-makers returned to the rest house and found that the gas stove was not connected to the gas cylinder. So, they became very sad, because it was not possible for them to connect such complex thing. And so many foods were about to be rotten. But luckily, they found the microwave oven working. So, they tried to boil all the eggs using the microwave oven (may be, first time in history)! And they succeeded to boil the eggs!

Now they have **n** eggs and a bowl. They put some eggs in the bowl with some water. And after that they put the bowl into the oven to boil the eggs. It's risky to put more than **P** eggs in the bowl and the bowl can carry at most **Q** gm of eggs. It takes **12** minutes to boil a bowl of eggs. Now you are given the weight of the eggs in gm, and the trouble-makers have exactly **12** minutes in their hand. You have to find the maximum number of eggs they can boil without taking any risk.

Input

The first line of input will contain **T (≤ 100)** denoting the number of cases.

Each case starts with **3** integers **n (1 ≤ n ≤ 30)**, **P (1 ≤ P ≤ 30)** and **Q (1 ≤ Q ≤ 30)**. The next line contains **n** positive integers (not greater than **10**) in non-descending order. These integers denote the weight of the eggs in gm.

Output

For each case, print the case number and the desired result.

Sample Input	Output for Sample Input

2
3 2 10
1 2 3
4 5 5
4 4 5 5
	Case 1: 2
	Case 2: 1

Problem Setter: Jane Alam Jan, Special Thanks: Sohel Hafiz

31. 11936 - The Lazy Lumberjacks

Time limit: 1.000 seconds

A: The Lazy Lumberjacks

Once upon a time in a far, far away forest, there was a team of lazy lumberjacks. Since they were too lazy to cut trees, they were always figuring out ways to sneak out of work. Their foreman, on the other side, was always trying to put them all to work.

After a lot of discussions the foreman and the lumberjacks came to an agreement: they will work, but only if the area of the forest assigned to each one was a triangle. If it was any other shape they will be free not to work that week. The idea was to give each lumberjack three numbers representing the length of each of the triangles side. If the numbers were correct and form a triangle, the lumberjacks had to work, else, they were free to leave and not work.

Since our lumberjacks are as cunning as they are lazy, they convince the foreman to let them determine the surface and the site in the forest were they will work. As a result, the lumberjacks keep passing the foreman sets of numbers that could not form the sides of a triangle. After a while, the foreman began to suspect and decide to write a program that validates the input of each lumberjack. Now when the lumberjacks decide to pass wrong numbers they get a fine of $1000.00 (more than a day's salary).

Your job is to write the program that the foreman has to use to determine if the numbers (all integers) passed by the lumberjacks can be the sides of a triangle. If they can, you have to print ``OK" else you have to print ``Wrong!!"

Input

The input consist in a data set describing the numbers of that each lumberjack has passed to the foreman for the day: The data is formatted as follows: The first line is an integer N ($2 \leq N \leq 20$). Then follows N lines, each one containing three integers separated by a space.

Output

For each line in the input you have to find if the integers can represent the sides of a triangle. If they can you have to print ``OK" for each line in the input, else you have to print ``Wrong!!"

Sample Input

```
6
1 2 3
3 2 5
3 4 5
6 6 1
3 3 3
7 3 10
```

Sample Output

```
Wrong!!
Wrong!!
OK
OK
OK
Wrong!!
```

32. 11942 - Lumberjack Sequencing

Time limit: 1.000 seconds

A: Lumberjack Sequencing

Another tale of lumberjacks?. Let see ...

The lumberjacks are rude, bearded workers, while foremen tend to be bossy and simpleminded. The foremen like to harass the lumberjacks by making them line up in groups of ten, ordered by the length of their beards. The lumberjacks, being of different physical heights, vary their arrangements to confuse the foremen. Therefore, the foremen must actually measure the beards in centimeters to see if everyone is lined up in order.

Your task is to write a program to assist the foremen in determining whether or not the lumberjacks are lined up properly, either from shortest to longest beard or from longest to shortest.

Input

The input starts with line containing a single integer N, $0 < N < 20$, which is the number of groups to process. Following this are N lines, each containing ten distinct positive integers less than 100.

Output

There is a title line, then one line per set of beard lengths. See the sample output for capitalization and punctuation.

Sample Input

```
3
13 25 39 40 55 62 68 77 88 95
88 62 77 20 40 10 99 56 45 36
91 78 61 59 54 49 43 33 26 18
```

Sample Output

```
Lumberjacks:
Ordered
Unordered
Ordered
```

33. 11984 - A Change in Thermal Unit

Time limit: 1.000 seconds

Measuring temperature and temperature differences are common task in many research and applications. Unfortunately, there exists more than one unit of measuring temperatures. This introduces a lot of confusion at times. Two popular units of measurements are Celsius(**C**) and Fahrenheit (**F**). The conversion of **F**from **C** is given by the formula:

$$\mathbf{F = \frac{9}{5} C + 32}$$

In this problem, you will be given an initial temperature in **C** and an increase in temperature in **F**. You would have to calculate the new temperature in **C**.

Input

Input starts with an integer **T (≤ 100)**, denoting the number of test cases.

Each case contains a line with two integers **C** and **d (0 ≤ C, d ≤ 100)**, where **C**represents the initial temperature in Celsius and **d** represents the increase in temperature in Fahrenheit.

Output

For each case, print the case number and the new temperature in Celsius after rounding it to two digits after the decimal point.

Sample Input	Output for Sample Input

`2`
`100 0`
`0 100`
	`Case 1: 100.00`
	`Case 2: 55.56`

Problem Setter: Shamim Hafiz, Special Thanks: Sohel Hafiz, Jane Alam Jan

39. 12372 - Packing for Holiday

Time limit: 3.000 seconds

Mr. Bean used to have a lot of problems packing his suitcase for holiday. So he is very careful for this coming holiday. He is more serious this time because he is going to meet his fiancée and he is also keeping frequent communication with you as a programmer friend to have suggestions. He gets confused when he buys a gift box for his fiancée because he can't decide whether it will fit in his suitcase or not. Sometimes a box doesn't fit in his suitcase in one orientation and after rotating the box to a different orientation it fits in the suitcase. This type of behavior makes him puzzled.

So to make things much simpler he bought another suitcase having same length, width and height, which is 20 inches. This measurement is taken from inside of the box. So a box which has length, width and height of 20 inches will just fit in this suitcase. He also decided to buy only rectangular shaped boxes and keep a measuring tape in his pocket. Whenever he chooses one gift box, which must be rectangular shaped, he quickly measures the length, width and height of the box. But still he can't decide whether it will fit in his suitcase or not. Now he needs your help. Please write a program for him which calculates whether a rectangular box fits in his suitcase or not provided the length, width and height of the box. Note that, sides of the box must be parallel to the sides of the suitcase.

Input

Input starts with an integer **T (T ≤ 100)**, which indicates the number of test cases.

Each of the next T line contains three integers **L**, **W** and **H (1 ≤ L, W, H ≤ 50)** denoting the length, width and height of a rectangular shaped box.

Output

For each test case, output a single line. If the box fits in the suitcase in any orientation having the sides of the box is parallel to the sides of the suitcase, this line will be"**Case #: good**", otherwise it will be "**Case #: bad**". In your output **#**will be replaced by the case number.

Please see the sample input and sample output for exact format.

Sample Input **Output for Sample Input**

2
20 20 20
1 2 21
	Case 1: good
	Case 2: bad

7. 10062 - Tell me the frequencies!

Time limit: 3.000 seconds

Problem H
Tell me the frequencies!
Input: standard input
Output: standard output

Given a line of text you will have to find out the frequencies of the ASCII characters present in it. The given lines will contain none of the first 32 or last 128 ASCII characters. Of course lines may end with '\n' and '\r' but always keep those out of consideration.

Input

Several lines of text are given as input. Each line of text is considered as a single input. Maximum length of each line is 1000.

Output

Print the ASCII value of the ASCII characters which are present and their frequency according to the given format below. A blank line should separate each set of output. Print the ASCII characters in the ascending order of their frequencies. If two characters are present the same time print the information of the ASCII character with higher ASCII value first. Input is terminated by end of file.

Sample Input:

```
AAABBC
122333
```

Sample Output:

```
67 1
66 2
65 3

49 1
50 2
51 3
```

Shahriar Manzoor
16-12-2000

19. 10222 - Decode the Mad man

Time limit: 3.000 seconds

Decode the Mad man

The Problem

Once in BUET, an old professor had gone completely mad. He started talking with some peculiar words. Nobody could realize his speech and lectures. Finally the BUET authority fall in great trouble. There was no way left to keep that man working in university. Suddenly a student (definitely he was a registered author at UVA ACM Chapter and hold a good rank on 24 hour-Online Judge) created a program that was able to decode that professor’s speech. After his invention, everyone got comfort again and that old teacher started his everyday works as before.

So, if you ever visit BUET and see a teacher talking with a microphone, which is connected to a IBM computer equipped with a voice recognition software and students are taking their lecture from the computer screen, don’t get thundered! Because now your job is to write the same program which can decode that mad teacher's speech!

The Input

The input file will contain only one test case i.e. the encoded message.

The test case consists of one or more words.

The Output

For the given test case, print a line containing the decoded words. However, it is not so hard task to replace each letter or punctuation symbol by the two immediately to its left alphabet on your standard keyboard.

Sample Input

```
k[r dyt I[o
```

Sample Output

```
how are you
```

Ahmed Shamsul Arefin

20. 10235 - Simply Emirp

Time limit: 3.000 seconds

Problem G: Simply Emirp

An integer greater than 1 is called a prime number if its only positive divisors (factors) are 1 and itself. Prime numbers have been studied over the years by a lot of mathematicians. Applications of prime numbers arise in Cryptography and Coding Theory among others.

Have you tried reversing a prime ? For most primes, you get a composite (43 becomes 34). *An Emirp (Prime spelt backwards) is a Prime that gives you a different Prime when its digits are reversed*. For example, 17 is *Emirp* because 17 as well as 71 are Prime. In this problem, you have to decide whether a number ***N*** is Non-prime or Prime or *Emirp*. Assume that ***1<□N<□1000000***.

Interestingly, Emirps are not new to NTU students. We have been boarding 199 and 179 buses for quite a long time!

Input

Input consists of several lines specifying values for ***N***.

Output

For each ***N*** given in the input, output should contain one of the following:

1. "***N*** `is not prime.`", if ***N*** is not a Prime number.
2. "***N*** `is prime.`", if ***N*** is Prime and ***N*** is not *Emirp*.
3. "***N*** `is emirp.`", if ***N*** is *Emirp*.

Sample Input

```
17
18
19
179
199
```

Sample Output

```
17 is emirp.
18 is not prime.
19 is prime.
179 is emirp.
199 is emirp.
```

22. 10252 - Common Permutation

Time limit: 3.000 seconds

Problem G
Common Permutation
Input: standard input
Output: standard output
Time Limit: 4 seconds
Memory Limit: 32 MB

Given two strings of lowercase letters, ***a*** and ***b***, print the longest string ***x*** of lowercase letters such that there is a permutation of ***x*** that is a subsequence of ***a*** and there is a permutation of ***x*** that is a subsequence of ***b***.

Input

Input file contains several lines of input. Consecutive two lines make a set of input. That means in the input file line **1** and **2** is a set of input, line **3** and **4** is a set of input and so on. The first line of a pair contains ***a*** and the second contains ***b***. Each string is on a separate line and consists of at most **1000** lowercase letters.

Output

For each set of input, output a line containing ***x***. If several ***x*** satisfy the criteria above, choose the first one in alphabetical order.

Sample Input:

```
pretty
women
walking
down
the
street
```

Sample Output:

```
e
nw
et
```

(World Finals Warm-up Contest, Problem Source: University of Alberta Local Contest)

23. 10260 - Soundex

Time limit: 3.000 seconds

Problem D: Soundex

Soundex coding groups together words that appear to sound alike based on their spelling. For example, "can" and "khawn", "con" and "gone" would be equivalent under Soundex coding.

Soundex coding involves translating each word into a series of digits in which each digit represents a letter:

```
      1 represents B, F, P, or V
      2 represents C, G, J, K, Q, S, X,  or Z
      3 represents D or T
      4 represents L
      5 represents M or N
      6 represents R
```

The letters A, E, I, O, U, H, W, and Y are not represented in Soundex coding, and repeated letters with the same code digit are represented by a single instance of that digit. Words with the same Soundex coding are considered equivalent.

Each line of input contains a single word, all upper case, less than 20 letters long. For each line of input, produce a line of output giving the Soundex code.

Sample Input

```
KHAWN
PFISTER
BOBBY
```

Output for Sample Input

```
25
1236
11
```

30. 10323 - Factorial! You Must be Kidding!!!

Time limit: 3.000 seconds

Arif has bought a super computer from Bongobazar. Bongobazar is a place in Dhakawhere second hand goods are found in plenty. So the super computer bought by him is also second hand and has some bugs. One of the bugs is that the range of unsigned long integer of this computer for **C/C++** compiler has changed. Now its new lower limit is **10000** and upper limit is **6227020800**. Arif writes a program in **C/C++** which determines the factorial of an integer. Factorial of an integer is defined recursively as:

Factorial (0) = 1
Factorial (n) = n*factorial (n-1).

Of course one can manipulate these expressions. For example, it can be written as
Factorial (n) =n*(n-1)*factorial (n-2)

This definition can also be converted to an iterative one.

But Arif knows that his program will not behave rightly in the super computer. You are to write program which will simulate that changed behavior in a Normal Computer.

Input
The input file contains several lines of input. Each line contains a single integer **n**. No integer has more than six digits. Input is terminated by end of file.

Output
For each line of input you should output a single line. This line will contain a single integer **n!** if the value of **n!** fits within the unsigned long integer of Arif's computer. Otherwise the line will contain one of the following two words

Overflow! //(When n! > 6227020800)
Underflow! //(When n! < 10000)

Sample Input

```
2
10
100
```

Sample Output

```
Underflow!
3628800
Overflow!
```

(The Decider Contest, Problem setter: Shahriar Manzoor)