03

SOME OF EASY UVA PROBLEMS

Set	: 03
Problem Type	: Ad Hoc
Problem Range	: 11000 – 12999
Total Problem	: 40

Xplosive
Nasif Ahmed
Email : nasif.002@gmail.com
Skype : nasif.02

INDEX OF PROBLEMS

1. 11044 - Searching for Nessy

Time limit: 3.000 seconds

Searching for Nessy

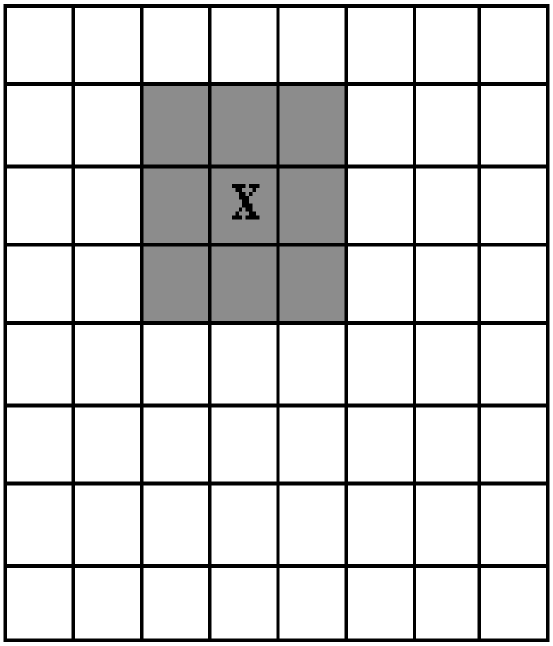
The Loch Ness Monsteris a mysterious and unidentified animal said to inhabit Loch Ness,
a large deep freshwater loch near the city of Inverness in northern Scotland. Nessie is usually categorized as a type of lake monster.
http://en.wikipedia.org/wiki/Loch_Ness_Monster

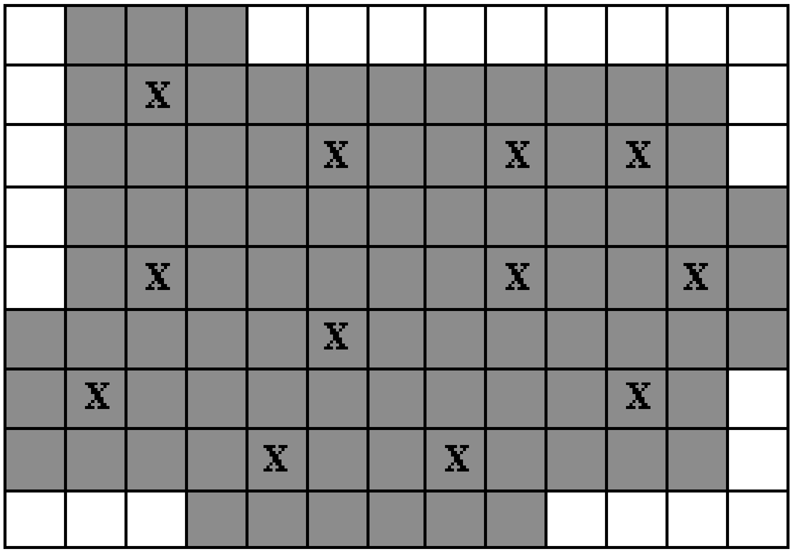
In July 2003, the BBC reported an extensive investigation of Loch Ness by a BBC team, using 600 separate sonar beams, found no trace of any ¨sea monster¨ (i.e., any large animal, known or unknown) in the loch. The BBC team concluded that Nessie does not exist. Now we want to repeat the experiment.

Given a grid of ***n*** rows and ***m*** columns representing the loch, 6 $\leq$ ***n***, ***m*** $\leq$ 10000, find the minimum number ***s*** of sonar beams you must put in the square such that we can control every position in the grid, with the following conditions:

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

For example,





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

Input

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

Output

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

Sample Input

```
3
6 6
7 7
9 13
```

Sample Output

```
4
4
12
```

2. 11058 - Encoding

Time limit: 3.000 seconds

You want to encode a string changing its letters. For each letter in the original string, where a letter is in the range **a-z**, you should replace it according to some encoding strategy. But that is not all! Your encoding strategy can also change and depends on the current position of the original string that you are analyzing.

In the beginning, there is an initial encoding strategy, specifying how you should replace the letters of the original string. At each step you can also have some additional rules, specifying a new enconding strategy for some letter. If there is not a new encoding strategy for a letter you should keep the old one. For example, we can estabilish that every **a** in the original string should be replaced by **b**, then you should estabilish that every **a** that appears since position **4** of the original string should be replaced by **c**.

Input

There are several test cases. Each test case starts with a string **S**, you can assume that the length of **S** is at most **100**. Then **26** lines will follow, specifying the encoding strategy for each letter, where the first line has the letter that should replace an **a** in the string **S**, the second line has the letter that should replace a **b** in the original string and so on.

After this, there is a line with an integer **0 ≤ R ≤ 1000** indicating the number of additional rules. Each one of the next **R** lines will have a rule, where each rule has the form **P X Y**, indicating that since position **P ≥ 0** of the original string you should replace the letter **X** by **Y**. The first letter of a string is at position **0**. You can assume that the initial position **P** of a rule in a test case is never less than the initial position of a previous rule in the same test case. If there are two rules for the same letter starting in the same position you should consider the rule that appears later in the input. There is a blank line after each test case and the input is finished by end of file (EOF).

Output

For each test case you must print the message: **Case #N: The encoding string is E.**, where **N** is the number of the test case, starting from **1**, and **E** is the string that you get after apply the set of encoding rules over the original string. After each test case you should print a blank line.

Sample Input

```
ufrn
t
o
w
k
q
z
f
n
y
i
c
m
s
j
n
r
g
l
d
s
u
s
g
y
e
u
10
0 q t
0 j f
1 v d
1 f d
1 r o
2 e p
2 v e
3 y p
3 t m
3 u k
```

Sample Output

```
Case #1: The encoding string is udoj.
```

3. 11121 - Base -2

Time limit: 3.000 seconds

Problem D
Base -2
Input: Standard Input

Output: Standard Output

> *The creator of the universe works in mysterious ways. But he uses a base ten counting system and likes round numbers.*

Scott Adams

Everyone knows about base-2 (binary) integers and base-10 (decimal) integers, but what about base **-2**? An integer n written in base -2 is a sequence of digits (b_i), writen right-to-left. Each of which is either 0 or 1 (no negative digits!), and the following equality must hold.

$n = b_0 + b_1(-2) + b_2(-2)^2 + b_3(-2)^3 + ...$

The cool thing is that every integer (including the negative ones) has a unique base**--2**representation, with no minus sign required. Your task is to find this representation.

Input

The first line of input gives the number of cases, **N** (at most 10000). **N** test cases follow. Each one is a line containing a decimal integer in the range from-1,000,000,000 to 1,000,000,000.

Output

For each test case, output one line containing "Case #**x**:" followed by the same integer, written in base -2 with no leading zeros.

Sample Input	Output for Sample Input
4 1 7 -2 0	Case #1: 1 Case #2: 11011 Case #3: 10 Case #4: 0

Problemsetter: Igor Naverniouk

Special Thanks: Shahriar Manzoor

4. 11152 - Colourful Flowers

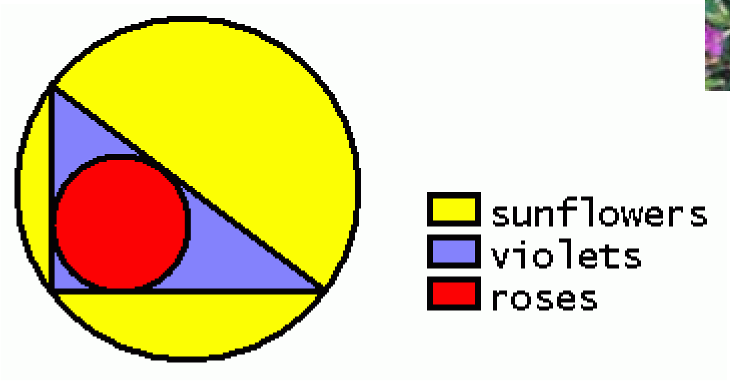
Time limit: 3.000 seconds

Problem E: Colourful Flowers

Time limit: 10 seconds

"Roses are red, violets are blue..."

Millionaire Mr Smith is well-known -- not for his wealth, but for his odd sense of "art"... Mr Smith has got a circular garden. On the boundary he picks three points and gets a triangle. He then finds the largest circle in that triangular region. So he gets something like this (Please click here for a black-and-white version of the figure):



Mr Smith then plants yellow sunflowers, blue violets and red roses in the way shown in the figure. (Nice combination, eh? :-) Given the lengths of the three sides of the triangle, you are to find the areas of the regions with each kind of flowers respectively.

Input and Output

Each line of input contains three integers a, b, c, the lengths of the three sides of the triangular region, with $0 < a \leq b \leq c \leq 1000$.

For each case, your program should output the areas of the regions with sunflowers, with violets and with roses respectively. Print your answers correct to 4 decimal places.

Sample Input

```
3 4 5
```

Sample Output

```
13.6350 2.8584 3.1416
```

Problemsetter: Mak Yan Kei

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	Output for Sample Input

3
10 20
20 10
10 10
	<
	
	>
	
	=

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	Output for Sample Input
10 100 1000 -1	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:

“**TOBENU**MBERON**EWEMEE**TAGAIN**ANDAGA**INUNDE**RBLUEI**CPCSKY”

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:

“UNEBOTNOREBM**EEMEWE**NIAGAT**AGADNA**EDNUNI**IEULBR**YKSCPC**”**

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`
`5 FA0ETASINAHGRI0NATWON0QA0NARI0`
`0`
	`CBASHEHSH`
	`ATE0AFGHANISTAN0IRAQ0NOW0IRAN0`

Problem-setter: Shahriar Manzoor

Special Thanks: Derek Kisman

8. 11219 - How old are you?

Time limit: 3.000 seconds

```
...
- Here are the filled form.
- Thank you. Let me check... hum... OK, OK, OK... Wait, how old are
you?
- 20. Did I forget to fill it?
- No. It says here that you'll be born next month! The year is
wrong...
- Oh... Sorry!
```

The process is going to be automatic and to avoid some human errors there will be a calculated field that informs the age based in the current date and the birth date given. This is your task, calculate the age, or say if there's something wrong.

Input

The first line of input gives the number of cases, T (1 ≤ T ≤ 200). T test cases follow. Each test case starts with a blank line, then you will have 2 lines corresponding to the current date and the birth date, respectively. The dates are in the format**DD**/**MM**/**YYYY**, where **DD** is the day, **MM** the month and **YYYY** the year. All dates will be valid.

Output

The output is comprised of one line for each input data set and should be as follow (quotes for clarifying only):

"Case #**N**: **AGE**", where **N** is the number of the current test case and **AGE** is one of the 3 following options:

- *"Invalid birth date"*, if the *calculated age* is impossible (still going to be born).
- *"Check birth date"*, if the *calculated age* is more than 130.
- the *calculated age* (years old only), otherwise.

Sample input

```
4

01/01/2007
10/02/2007

09/06/2007
28/02/1871

12/11/2007
01/01/1984

28/02/2005
29/02/2004
```

Sample output

```
Case #1: Invalid birth date
Case #2: Check birth date
Case #3: 23
Case #4: 0
```

9. 11220 - Decoding the message.

Time limit: 3.000 seconds

Chico and Maria are relatives that live in different towns. As they inhabit a rural area, it is very difficult for them to keep in touch. One way they found to overcome their communication problem was to send a line through their parents that used to visit each other.

The point is that Chico and Maria did not want that their parents read their messages, and they decided to create a secret code for the messages. The code is not very sophisticated, but you should keep in mind Chico and Maria are just children.

In general, the meaning of a message is based on a letter of each word, in a way that they will form a message with the first letter of the first word, the second letter of the second word and so on. If a word does not have enough letters, the following word should be used. For example, if you are analyzing the third word, you should consider its third letter, but if it just has two letters, then you should try to form a decoded word with the third letter of the fourth word.

When the end of a line is reached, you should finish the current decoded word and should start to form another one from the first letter of the first word in the next line.

Your task is to translate a message according to Chico and Maria's secret code.

Input

The first line of input gives the number of cases, T (1 ≤ T ≤ 30), then there is a blank line before the first test case. Each test case represents a message, which is composed by 1 ≤ N ≤ 100 lines and each line is composed by 1 ≤ M ≤ 30 words. Two words in the same line are separated by one or more white spaces. A word is formed by the letters **A-Z** and **a-z** and has at most 30 letters. The only symbols that appear in the input are the alphabetic letters and white spaces. There will be a blank line after each message.

Output

For each test case you must print the number of the test case and each word of the decoded message, one per line (look the sample output for the exact format). You must print a blank line between each test case.

Sample Input

```
2

Hey good lawyer
as I previously previewed
yam does a soup

First I give money to Teresa
after I inform dad of
your horrible soup
```

Sample Output

```
Case #1:
How
are
you

Case #2:
Fine
and
you
```

10. 11223 - O: dah dah dah!

Time limit: 3.000 seconds

E. O: dah, dah, dah!

Time limit: 1s

Morse code is a method for long-distance transmission of textual information without using the usual symbols. Instead information is represented with a simpler, binary, alphabet composed of short and long beeps. The short beep is called dih, and the long beep is called dah. For instance, the code for the letter O is dah dah dah (three long beeps). Actually, because the codification is not prefix-free, there is also a third symbol, which is silence. The code between two letters is a simple silence, the code between two words is a double silence.

You have been assigned the job to translate a message in Morse code. The signal has already been digitalized in the following fashion: dih is represented by a dot (.), dah is represented by a dash (-). Simple and double silences are represented by a single space character and two space characters respectively.

The following table represents the Morse code of all the characters that your program need to be able to handle.

Symbol	Code	Symbol	Code	Symbol	Code	Symbol	Code	Symbol	Code	Symbol	Code
A	.-	J	.- - -	S	...	1	.- - - -	.	.-.-.-	:	- - -...
B	-...	K	-.-	T	-	2	..- - -	,	- -..- -	;	-.-.-.
C	-.-.	L	.-..	U	..-	3	...- -	?	..- -..	=	-...-
D	-..	M	- -	V	...-	4	....-	'	.- - - -.	+	.-.-.
E	.	N	-.	W	.- -	5	.....	!	-.-.- -	-	-....-
F	..-.	O	- - -	X	-..-	6	-....	/	-..-.	_	..- -.-
G	- -.	P	.- -.	Y	-.- -	7	- -...	(	-.- -.	"	.-..-.
H	....	Q	- -.-	Z	- -..	8	- - -..	)	-.- -.-	@	.- -.-.
I	..	R	.-.	0	- - - - -	9	- - - -.	&	.-...		

Input

The first line of input gives the number of cases, T (1 ≤ T ≤ 10). T test cases follow. Each one is a sequence of dot, dash and space characters. Two messages are separated by a newline. The maximum length of a message is 2000.

Output

The output is comprised of one paragraph for each message. The paragraph corresponding to the n-th message starts with the header `Message #n`, on a line on its own. Each decoded sentence of the message appears then successively on a line of its own. Two paragraphs are separated by a blank line. The sentences shall be printed in uppercase.

Sample input

```
2
... --- ...
.--- --- -...  -.. --- -. .  ..--..  ..-. .. -. . -.-.--
```

Sample output

```
Message #1
SOS

Message #2
JOB DONE ? FINE!
```

Problem setter: David Deharbe
Universidade Federal do Rio Grande do Norte Qualifying Contest IV, June 9th, 2007.

11. 11233 - Deli Deli

Time limit: 3.000 seconds

Mrs. Deli is running the delicatessen store "Deli Deli". Last year Mrs. Deli has decided to expand her business and build up an online store. She has hired a programmer who has implemented the online store.

Recently some of her new online customers complained about the electronic bills. The programmer had forgotten to use the plural form in case that an item is purchased multiple times. Unfortunaly the programmer of Mrs. Deli is on holiday and now it is your task to implement this feature for Mrs. Deli. Here is a description how to make the plural form:

1. If the word is in the list of irregular words replace it with the given plural.
2. Else if the word ends in a consonant followed by "y", replace "y" with "ies".
3. Else if the word ends in "o", "s", "ch", "sh" or "x", append "es" to the word.
4. Else append "s" to the word.

Input Specification

The first line of the input consists of two integers **L** and **N** ($0 \leq L \leq 20, 1 \leq N \leq 100$). The following **L** lines contain the description of the irregular words and their plural form. Each line consists of two words separated by a space character, where the first word is the singular, the second word the plural form of some irregular word. After the list of irregular words, the following **N** lines contain one word each, which you have to make plural. You may assume that each word consists of at most 20 lowercase letters from the english alphabet ('a' to 'z').

Output Specification

Print **N** lines of output, where the **ith** line is the plural form of the **ith** input word.

Sample Input

```
3 7
rice rice
spaghetti spaghetti
octopus octopi
rice
lobster
spaghetti
strawberry
octopus
peach
turkey
```

Sample Output

```
rice
lobsters
spaghetti
strawberries
octopi
peaches
turkeys
```

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

13. 11340 - Newspaper

Time limit: 1.000 seconds

A - Newspaper

Time Limit: 1 sec
Memory Limit: 16MB

News agency pays money for articles according to some rules. Each character has its own value (some characters may have value equals to zero). Author gets his payment as a sum of all character's values in the article. You have to determine the amount of money that news agency must pay to an author.

INPUT:

The first line contains integer N (0 < N <= 5), it is a number of tests. Each test describes an integer K (0 < K <= 100), the number of paid characters. On next K lines there are table of paid characters and its values (character values are written in cents). If character can not be found in the table, then its value is equal to zero. Next, there is integer M (0 < M <= 150000). Next M lines contain an article itself. Each line can be up to 10000 characters length. Be aware of a large input size, the whole input file is about 7MB.

OUTPUT:

For each test print how much money publisher must pay for an article in format "`x.yy$`". Where "`x`" is a number of dollars without leading zeros, and "`yy`" number of cents with one leading zero if necessary. Examples: "`3.32$`", "`13.07$`", "`71.30$`", "`0.09$`".

SAMPLE INPUT:

```
1
7
a 3
W 10
A 100
, 10
k 7
. 3
I 13
7
ACM International Collegiate Programming Contest (abbreviated
as ACM-ICPC or just ICPC) is an annual multi-tiered competition
among the universities of the world. The ICPC challenges students
to set ever higher standards of excellence for themselves
through competition that rewards team work, problem analysis,
and rapid software development.
From Wikipedia.
```

SAMPLE OUTPUT:

```
3.74$
```

Problem setters: Aleksej Viktorchik, Leonid Shishlo.
Huge Easy Contest #1

14. 11343 - Isolated Segments

Time limit: 1.000 seconds

You're given n segments in the rectangular coordinate system. The segments are defined by start and end points (X_i,Y_i) and (X_j,Y_j) (1 <= i, j <= n). Coordinates of these points are integer numbers with real value smaller then 1000. Length of each segment is positive.

When 2 segments don't have a common point then it is said that segments don't collide. In any other case segments collide. Be aware that segments collide even if they have only one point in common.

Segment is said to be isolated if it doesn't collide with all the other segments that are given, i.e. segment i is isolated when for each 1 <= j <= n, (i != j), segments i and j don`t collide. You are asked to find number T - how many segments are isolated.

INPUT:

First line of input contains number N (N <= 50), then tests follow. First line of each test case contains number M (M <= 100) - the number of segments for this test case to be considered. For this particular test case M lines follow each containing a description of one segment. Segment is described by 2 points: start point (X_{pi},Y_{pi}) and end point (X_{ei},Y_{ei}). They are given in such order: X_{pi} Y_{pi} X_{ei} Y_{ei}

OUTPUT:

For each test case output one line containing number T.

SAMPLE INPUT:

```
6
3
0 0 2 0
1 -1 1 1
2 2 3 3
2
0 0 1 1
1 0 0 1
2
0 0 0 1
0 2 0 3
2
0 0 1 0
1 0 2 0
2
0 0 2 2
1 0 1 1
2
1 3 1 5
1 0 1 6
```

SAMPLE OUTPUT:

```
1
0
2
0
0
0
```

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	

17. 11437 - Triangle Fun

Time limit: 1.000 seconds

I. Behold My Quadrangle

Context

Any square is a rectangle, any rectangle is a quadrangle, and any quadrangle is composed of four sides. But not all rectangles are squares, not all quadrangles are rectangles, and not all sets of four sides are quadrangles.

The Problem

We have the length of four sides. You have to determine if they can form a square. If not, determine if they can form a rectangle. If not, determine if they can form a quadrangle.

The Input

The first line of the input contains an integer indicating the number of test cases.

For each test case, there is a line with four positive integer numbers, between 0 and 2^{30}.

The Output

For each test case, the output should consist of a line with the text "`square`", "`rectangle`", "`quadrangle`" or "`banana`", if the sides of the corresponding case can form a square, a rectangle, a quadrangle or none, respectively.

Sample Input

```
4
10 8 7 6
9 1 9 1
29 29 29 29
5 12 30 7
```

Sample Output

```
quadrangle
rectangle
square
banana
```

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

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

Problem Setter: Mohammad Mahmudur Rahman

Special Thanks: Shahriar Manzoor

19. 11498 - Division of Nlogonia

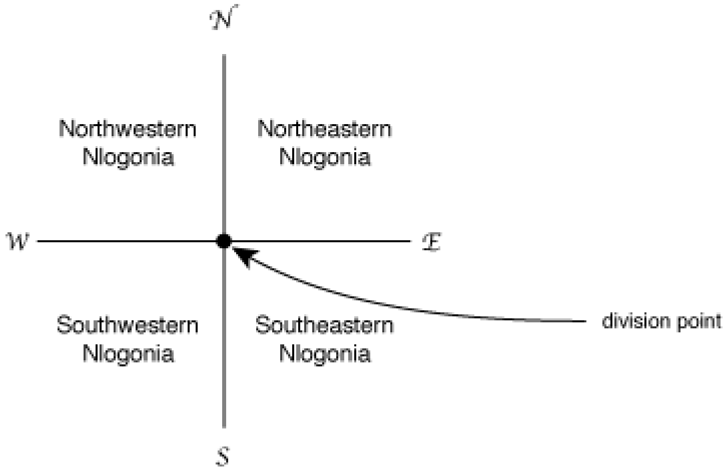
Time limit: 1.000 seconds

Division of Nlogonia

The Problem

After centuries of hostilities and skirmishes between the four nations living in the land generally known as Nlogonia, and years of negotiations involving diplomats, politicians and the armed forces of all interested parties, with mediation by UN, NATO, G7 and SBC, it was at last agreed by all the way to end the dispute, dividing the land into four independent territories.

It was agreed that one point, called *division point*, with coordinates established in the negotiations, would define the country division, in the following way. Two lines, both containing the division point, one in the North-South direction and one in the East-West direction, would be drawn on the map, dividing the land into four new countries. Starting from the Western-most, Northern-most quadrant, in clockwise direction, the new countries will be called Northwestern Nlogonia, Northeastern Nlogonia, Southeastern Nlogonia and Southwestern Nlogonia.



The UN determined that a page in the Internet should exist so that the inhabitants could check in which of the countries their homes are. You have been hired to help implementing the system.

The Input

The input contains several test cases. The first line of a test case contains one integer *K*indicating the number of queries that will be made ($0 < K \leq 10^3$). The second line of a test case contains two integers *N* and *M* representing the coordinates of the division point ($-10^4 < N, M < 10^4$). Each of the *K* following lines contains two integers *X* and*Y* representing the coordinates of a residence ($-10^4 \leq X, Y \leq 10^4$).

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

The Output

For each test case in the input your program must print one line containing:

- the word `divisa` (means border in Portuguese) if the residence is on one of the border lines (North-South or East-West);
- NO (means NW in Portuguese) if the residence is in Northwestern Nlogonia;
- NE if the residence is in Northeastern Nlogonia;
- SE if the residence is in Southeastern Nlogonia;
- SO (means SW in Portuguese) if the residence is in Southwestern Nlogonia.

Sample Input

```
3
2 1
10 10
-10 1
0 33
4
-1000 -1000
-1000 -1000
0 0
-2000 -10000
-999 -1001
0
```

Sample Output

```
NE
divisa
NO
divisa
NE
SO
SE
```

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	

21. 11547 - Automatic Answer

Time limit: 1.000 seconds

Last month Alice nonchalantly entered her name in a draw for a Tapmaster 4000. Upon checking her mail today, she found a letter that read:

“Congratulations, Alice! You have won a Tapmaster 4000. To claim your prize, you must answer the following skill testing question.”

Alice’s initial feelings of surprised joy turned quickly to those of dismay. Her lifetime record for skill testing questions is an abysmal 3 right and 42 wrong.

Mad Skills, the leading skill testing question development company, was hired to provide skill testing questions for this particular Tapmaster 4000 draw. They decided to create a different skill testing question to each winner so that the winners could not collaborate to answer the question.

Can you help Alice win the Tapmaster 4000 by solving the skill testing question?

Program Input

The input begins with t ($1 \leq t \leq 100$), the number of test cases. Each test case contains an integer n ($-1000 \leq n \leq 1000$) on a line by itself. This n should be substituted into the skill testing question below.

Program Output

For each test case, output the answer to the following skill testing question on a line by itself: “Multiply n by 567, then divide the result by 9, then add 7492, then multiply by 235, then divide by 47, then subtract 498. What is the digit in the tens column?”

Sample Input & Output

INPUT

```
2
637
-120
```

OUTPUT

```
1
3
```

Calgary Collegiate Programming Contest 2008

22. 11616 - Roman Numerals

Time limit: 1.000 seconds

Context

This year is the XXV Anniversary of the Faculty of Computer Science in Murcia. But, what XXV means? Maybe you should ask an ancient Roman to get the answer.

The Problem

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

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

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

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

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

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

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

The Input

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

The Output

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

Sample Input

```
XXV
4
942
MCMLXXXIII
```

Sample Output

```
25
IV
CMXLII
1983
```

23. 11727 - Cost Cutting

Time limit: 1.000 seconds

Company XYZ have been badly hit by recession and is taking a lot of cost cutting measures. Some of these measures include giving up office space, going open source, reducing incentives, cutting on luxuries and issuing pink slips.

They have got three (3) employees working in the accounts department and are going to lay-off two (2) of them. After a series of meetings, they have decided to dislodge the person who gets the most salary and the one who gets the least. This is usually the general trend during crisis like this.

You will be given the salaries of these 3 employees working in the accounts department. You have to find out the salary of the person who survives.

Input

The first line of input is an integer **T (T<20)** that indicates the number of test cases. Each case consists of a line with 3 distinct positive integers. These 3 integers represent the salaries of the three employees. All these integers will be in the range [1000, 10000].

Output

For each case, output the case number followed by the salary of the person who survives.

Sample Input

```
3
1000 2000 3000
3000 2500 1500
1500 1200 1800
```

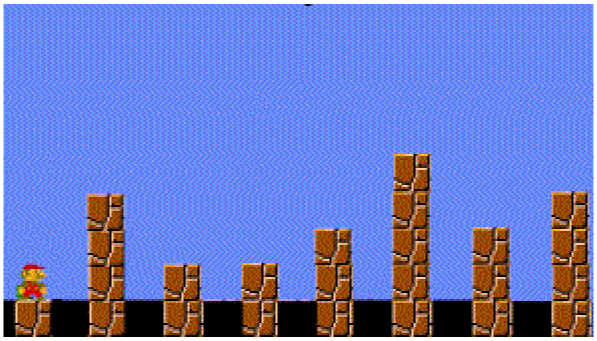
Output for Sample Input

```
Case            1:            2000
Case            2:            2500
Case 3: 1500
```

Problem Setter: Sohel Hafiz, Special Thanks: Md. Arifuzzaman Arif

24. 11764 - Jumping Mario

Time limit: 3.000 seconds



Mario is in the final castle. He now needs to jump over few walls and then enter the Koopa's Chamber where he has to defeat the monster in order to save the princess. For this problem, we are only concerned with the "jumping over the wall" part. You will be given the heights of **N**walls from left to right. Mario is currently standing on the first wall. He has to jump to the adjacent walls one after another until he reaches the last one. That means, he will make (**N-1**) jumps. A *high jump* is one where Mario has to jump to a taller wall, and similarly, a low jump is one where Mario has to jump to a shorter wall. Can you find out the total number of *high jumps* and *low jumps* Mario has to make?

Input

The first line of input is an integer **T** (**T < 30**) that indicates the number of test cases. Each case starts with an integer **N** (**0 < N < 50**) that determines the number of walls. The next line gives the height of the **N** walls from left to right. Each height is a positive integer not exceeding 10.

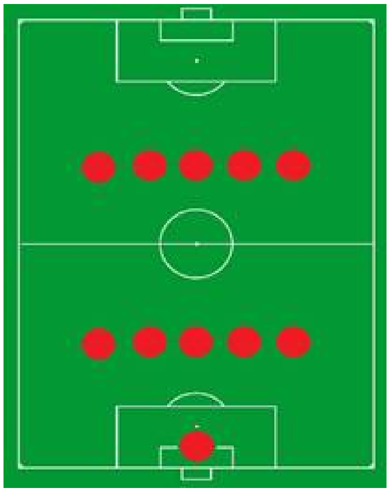
Output

For each case, output the case number followed by 2 integers, total high jumps and total low jumps, respectively. Look at the sample for exact format.

Sample Input	**Output for Sample Input**
3	Case 1: 4 2
8	Case 2: 0 0
1 4 2 2 3 5 3 4	Case 3: 4 0
1	
9	
5	
1 2 3 4 5	

25. 11804 - Argentina

Time limit: 1.000 seconds



The Argentine football team coach, the great Diego Maradona, is going to try out a new*formation* this year. *Formation* describes how the players are positioned on the pitch. Instead of the conventional 4-4-2 or 4-3-3, he has opted for 5-5. This means there are 5 attackers and 5 defenders.

You have been hired by *Argentina Football Federation* (AFF) to write a code that will help them figure out which players should take the attacking/defensive positions.

Maradona has given you a list containing the names of the 10 players who will take the field. The attacking ability and the defensive ability of each player are also given. Your job is to figure out which 5 players should take the attacking positions and which 5 should take the defensive positions.

The rules that need to be followed to make the decision are:

- The sum of the attacking abilities of the 5 attackers needs to be maximized
- If there is more than one combination, maximize the sum of the defending abilities of the 5 defenders
- If there is still more than one combination, pick the attackers that come lexicographically earliest.

Input

The first line of input contains an integer **T**(**T**<50) that indicates the number of test cases. Each case contains exactly 10 lines. The **i**$^{\mathbf{th}}$ line contains the name of the **i**$^{\mathbf{th}}$player followed by the attacking and defending ability of that player respectively. The length of a player's name is at most 20 and consists of lowercase letters only. The attacking/defending abilities are integers in the range [0, 99].

Output

The output of each case contains three lines. The first line is the case number starting from 1. The next line contains the name of the 5 attackers in the format (A_1, A_2, A_3, A_4, A_5) where A_i is the name of an attacker. The next line contains the name of the 5 defenders in the same format. The attackers and defenders names should be printed in lexicographically ascending order. Look at the sample for more details.

Sample Input	Sample Output
1 sameezahur 20 21 sohelh 18 9 jaan 17 86 sidky 16 36 shamim 16 18 shadowcoder 12 9 muntasir 13 4 brokenarrow 16 16 emotionalblind 16 12 tanaeem 20 97	Case 1: (emotionalblind, jaan, sameezahur, sohelh, tanaeem) (brokenarrow, muntasir, shadowcoder, shamim, sidky)

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**

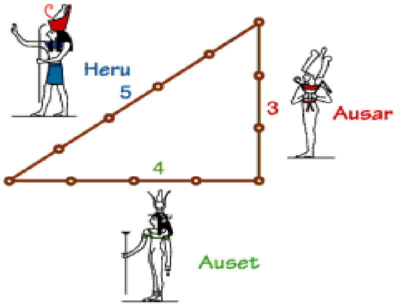
3
10 20 30 40
7 5 12
125 15 25
	20
	1
	25

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

29. 11917 - Do Your Own Homework

Time limit: 1.000 seconds

These days Soha is so busy that he doesn't have time to do his own homework. But this is not a big problem since he has got many friends who are willing to help. One if his friend's name is Sparrow. Whenever Soha is assigned any homework, he turns to Sparrow for her help.

Sparrow has given a list of subjects that she is comfortable with along with the number of days it will take her to complete an assignment for each subject. Soha has got only*D* days to complete his next assignment. However, the professor of this subject is a little flexible and allows late submissions up to 5 days. That means he will not accept any submission that is after $D + 5$ days from now. Will Sparrow be able to do it for Soha this time?

Input

First line of input is a positive integer *T* (*T le*100) that determines the number of test cases. Each case starts with a line containing an integer *N* that represents the number of subjects Sparrow is comfortable with. Each of the next *N* lines contain the name of a subject followed by the number of days it will take Sparrow to complete an assignment of that subject. All these subject names will be distinct. The next line contains an integer*D*. The meaning of *D* is described above. The following line contains the name of the subject whose homework is due. All the subjects' names consist of lowercase letters and the length of each is at least 1 and at most 20. All the integer inputs are positive in the range [1,100].

Output

For each case, first output the case number first starting from 1. If Sparrow doesn't take more than *D* days to completely the assignment, output `Yesss'; if she takes more than *D* days but not more than $D + 5$, output `Late'; if she takes more than $D + 5$ days or if she isn't comfortable with the subject, output `Do your own homework!'. Quotes are for clarify only and don't need to be part of the output. Look at the samples for more details. Be careful about the spelling.

Sample Input

```
3
3
compiler 4
cplusplus 1
java 8
5
compiler
2
algorithm 3
math 9
4
math
2
java 8
ai 3
6
calculus
```

Sample Output

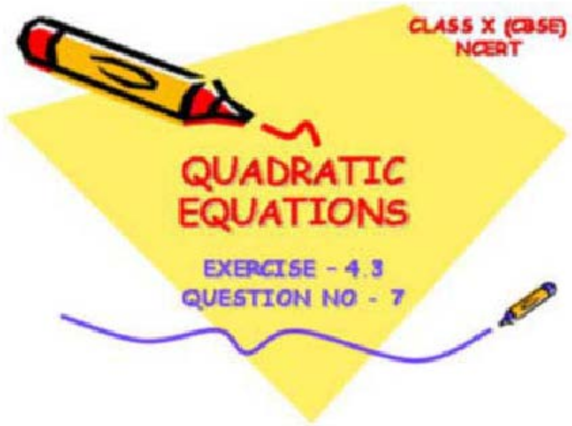
```
Case 1: Yesss
Case 2: Late
Case 3: Do your own homework!
```

30. 11934 - Magic Formula

Time limit: 1.000 seconds

Magic Formula

You are given a quadratic function, $f(n) = a \times 2 + b \times + c$. You are also given a divisor d and a limit L. How many of the function values $f(0), f(1), \ldots, f(L)$ are divisible by d?



Input

Input consists of a number of test cases. Each test case consists of a single line containing the numbers a b c d L ($-1000 \leq a, b, c \leq 1000$, $1 < d < 1000000$, $0 \leq L < 1000$).

Input is terminated by a line containing `0 0 0 0 0' which should not be processed.

Output

Print the answer for each test case (the number of function values $f(0), f(1), \ldots, f(L)$divisible by d) on a separate line.

Sample Input

```
0 0 10 5 100
0 0 10 6 100
1 2 3 4 5
1 2 3 3 5
0 0 0 0 0
```

Sample Output

```
101
0
0
4
```

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

34. 11988 - Broken Keyboard (a.k.a. Beiju Text)

Time limit: 1.000 seconds

Problem B

Broken Keyboard (a.k.a. Beiju Text)

You're typing a long text with a broken keyboard. Well it's not so badly broken. The only problem with the keyboard is that sometimes the "home" key or the "end" key gets automatically pressed (internally).

You're not aware of this issue, since you're focusing on the text and did not even turn on the monitor! After you finished typing, you can see a text on the screen (if you turn on the monitor).

In Chinese, we can call it Beiju. Your task is to find the Beiju text.

Input

There are several test cases. Each test case is a single line containing at least one and at most 100,000 letters, underscores and two special characters '[' and ']'. '[' means the "Home" key is pressed internally, and ']' means the "End" key is pressed internally. The input is terminated by end-of-file (EOF). The size of input file does not exceed 5MB.

Output

For each case, print the Beiju text on the screen.

Sample Input

```
This_is_a_[Beiju]_text
[[]][][]Happy_Birthday_to_Tsinghua_University
```

Output for the Sample Input

```
BeijuThis_is_a__text
Happy_Birthday_to_Tsinghua_University
```

Rujia Liu's Present 3: A Data Structure Contest Celebrating the 100th Anniversary of Tsinghua University

Special Thanks: Yiming Li

Note: Please make sure to test your program with the gift I/O files before submitting!

35. 11991 - Easy Problem from Rujia Liu?

Time limit: 1.000 seconds

Problem E

Easy Problem from Rujia Liu?

Though Rujia Liu usually sets hard problems for contests (for example, regional contests like Xi'an 2006, Beijing 2007 and Wuhan 2009, or UVa OJ contests like Rujia Liu's Presents 1 and 2), he occasionally sets easy problem (for example, 'the Coco-Cola Store' in UVa OJ), to encourage more people to solve his problems :D

Given an array, your task is to find the k-th occurrence (from left to right) of an integer v. To make the problem more difficult (and interesting!), you'll have to answer m such queries.

Input

There are several test cases. The first line of each test case contains two integers n, m(1<=n,m<=100,000), the number of elements in the array, and the number of queries. The next line contains n positive integers not larger than 1,000,000. Each of the following m lines contains two integer k and v (1<=k<=n, 1<=v<=1,000,000). The input is terminated by end-of-file (EOF). The size of input file does not exceed 5MB.

Output

For each query, print the 1-based location of the occurrence. If there is no such element, output 0 instead.

Sample Input

```
8 4
1 3 2 2 4 3 2 1
1 3
2 4
3 2
4 2
```

Output for the Sample Input

```
2
0
7
0
```

Rujia Liu's Present 3: A Data Structure Contest Celebrating the 100th Anniversary of Tsinghua University

Special Thanks: Yiming Li

Note: Please make sure to test your program with the gift I/O files before submitting!

36. 11995 - I Can Guess the Data Structure!

Time limit: 1.000 seconds

There is a bag-like data structure, supporting two operations:

```
1 x
```

Throw an element x into the bag.

```
2
```

Take out an element from the bag.

Given a sequence of operations with return values, you're going to guess the data structure. It is a stack (Last-In, First-Out), a queue (First-In, First-Out), a priority-queue (Always take out larger elements first) or something else that you can hardly imagine!

Input

There are several test cases. Each test case begins with a line containing a single integer n (1<=n<=1000). Each of the next n lines is either a type-1 command, or an integer 2 followed by an integer x. That means after executing a type-2 command, we get an element x *without error*. The value of x is always a positive integer not larger than 100. The input is terminated by end-of-file (EOF). The size of input file does not exceed 1MB.

Output

For each test case, output one of the following:

```
stack
```

It's definitely a stack.

```
queue
```

It's definitely a queue.

```
priority queue
```

It's definitely a priority queue.

```
impossible
```

It can't be a stack, a queue or a priority queue.

```
not sure
```

It can be more than one of the three data structures mentioned above.

Sample Input

```
6
```

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

Output for the Sample Input

```
queue
not sure
impossible
stack
priority queue
```

Rujia Liu's Present 3: A Data Structure Contest Celebrating the 100th Anniversary of Tsinghua University

Special Thanks: Yiming Li

Note: Please make sure to test your program with the gift I/O files before submitting!

37. 12015 - Google is Feeling Lucky

Time limit: 1.000 seconds

Google is one of the most famous Internet search engines which hosts and develops a number of Internet-based services and products. On its search engine website, an interesting button `I'm feeling lucky' attracts our eyes. This feature could allow the user skip the search result page and goes directly to the first ranked page. Amazing! It saves a lot of time.

The question is, when one types some keywords and presses `I'm feeling lucky' button, which web page will appear? Google does a lot and comes up with excellent approaches to deal with it. In this simplified problem, let us just consider that Google assigns every web page an integer-valued relevance. The most related page will be chosen. If there is a tie, all the pages with the highest relevance are possible to be chosen.

Your task is simple, given 10 web pages and their relevance. Just pick out all the possible candidates which will be served to the user when `I'm feeling lucky'.

Input

The input contains multiple test cases. The number of test cases *T* is in the first line of the input file.

For each test case, there are 10 lines, describing the web page and the relevance. Each line contains a character string without any blank characters denoting the URL of this web page and an integer V_i denoting the relevance of this web page. The length of the URL is between 1 and 100 inclusively. ($1 \leq V_i \leq 100$)

Output

For each test case, output several lines which are the URLs of the web pages which are possible to be chosen. The order of the URLs is the same as the input. Please look at the sample output for further information of output format.

Sample Input

```
2
www.youtube.com 1
www.google.com 2
www.google.com.hk 3
www.alibaba.com 10
www.taobao.com 5
www.bad.com 10
www.good.com 7
www.fudan.edu.cn 8
www.university.edu.cn 9
acm.university.edu.cn 10
www.youtube.com 1
www.google.com 2
www.google.com.hk 3
www.alibaba.com 11
www.taobao.com 5
www.bad.com 10
www.good.com 7
www.fudan.edu.cn 8
acm.university.edu.cn 9
acm.university.edu.cn 10
```

Sample Output

```
Case #1:
www.alibaba.com
www.bad.com
acm.university.edu.cn
Case #2:
www.alibaba.com
```

38. 12149 - Feynman

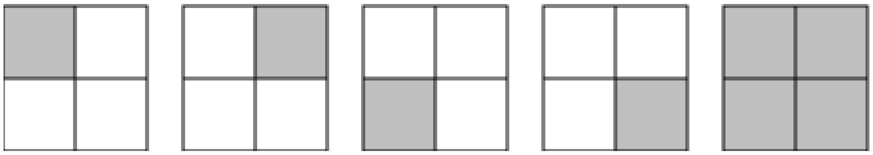
Time limit: 3.000 seconds

Feynman

Richard Phillips Feynman was a well known American physicist and a recipient of the Nobel Prize in Physics. He worked in theoretical physics and also pioneered the field of quantum computing. He visited South America for ten months, giving lectures and enjoying life in the tropics. He is also known for his books "Surely You're Joking, Mr. Feynman!" and "What Do You Care What Other People Think?", which include some of his adventures below the equator.

His life-long addiction was solving and making puzzles, locks, and cyphers. Recently, an old farmer in South America, who was a host to the young physicist in 1949, found some papers and notes that is believed to have belonged to Feynman. Among notes about mesons and electromagnetism, there was a napkin where he wrote a simple puzzle: "how many different squares are there in a grid of $N \times N$ squares?".

In the same napkin there was a drawing which is reproduced below, showing that, for$N$=2, the answer is 5.



Input

The input contains several test cases. Each test case is composed of a single line, containing only one integer N, representing the number of squares in each side of the grid ($1 \le N \le 100$).

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

Output

For each test case in the input, your program must print a single line, containing the number of different squares for the corresponding input.

Sample input	**Output for the sample input**
2 1 8 0	5 1 204

ACM ICPC :: South American Regional 2008

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

```
2

20 20 20

1 2 21
```

Output for Sample Input

```
Case 1: good

Case 2: bad
```

40. 12602 - Nice Licence Plates

Time limit: 1.000 seconds

Problem A: Nice Licence Plates

Alberta licence plates currently have a format of ABC-0123 (three letters followed by four digits).

We say that the licence plate is "nice" if the absolute difference between the value of the first part and the value of the second part is at most 100.

The value of the first part is calculated as the value of base-26 number (where digits are in [A..Z]). For instance, if the first part is "ABC", its value is 28 (0*26^2 + 1*26^1 + 2*26^0). So, the plate "ABC-0123" is nice, because |28-123|<=100.

Given the list of licence plate numbers, your program should determine if the plate is nice or not.

Input Format

First line of the input contains an integer N (1<=N<=100), the number of licence plate numbers. Then follow N lines, each containing a licence plate in the format LLL-DDDD.

Output Format

For each licence plate print on a line "nice" or "not nice" (without quotes) depending on the plate number being nice as described in the probem statement.

Sample Input

```
2
ABC-0123
AAA-9999
```

Sample Output

```
nice
not nice
```

Darko Aleksic

ACPC 2012