45^{th}
ACM LOCAL PROGRAMMING CONTEST

Isfahan University of Technology

Problem		
	Name	Color
A	Oogway's Army	Red
B	Oogway's Basic Problem of Kung Fu	Orange
C	Mr. Ping and His Odd Hobby	Yellow
D	Po and The Damn Stairs of Jade Palace	Light Green
E	A Panda Can't Fly!	Purple
F	Dragon Warrior's Tower	Dark Green
G	Po's Money	Light Pink
H	Everybody Knows Kung Fu Fighting!	Light Blue
I	Mr. Ping's Farm	White
J	Po's Money	Dark Pink
K	Final Words in Oogway's Library	Dark Blue
L	Happy Ending in the Valley of Peace	Brown

Problem A: Oogway's Army

Master Oogway gathered a large army in the Valley of Peace, and each day Lord Shen attacks with his army to the valley. But both sides are filled with tired and busy soldiers. Each day k warriors in the Jade Palace are awake and not busy. On the other side in Shen's army there are m warriors healthy and ready to fight. Our Spies are counting both sides and each day they will give us the number of warriors in both sides. Your task is to count the difference of Armies in each day.

Input

The input contains two numbers in every line. These two numbers in each line denotes the number soldiers in Oogway's army and his opponent's army or vice versa. The input numbers are not greater than 2^{32}. Input is terminated by 'End of File'.

Output

For each line of input, print the difference of number of soldiers between Oogways's army and his opponent's army. Each output should be in seperate line.

Sample Input

```
10 12
10 14
100 200
```

Sample Output

```
2
4
100
```

Problem B: Oogway's Basic Problem of Kung Fu

Master Oogway reached a new secret in the Universe. There were all numbers creating the power of Wuxi Finger Hold but not all numbers. The numbers should generate palindromes after doing some magics.

The 'reverse and add' method is simple: choose a number, reverse its digits and add it to the original. If the sum is not a palindrome (which means, it is not the same number from left to right and right to left), repeat this procedure.

In this particular case the palindrome '9339' appeared after the 4th addition. This method leads to palindromes in a few step for almost all of the integers. But there are interesting exceptions. 196 is the first number for which no palindrome has been found. It is not proven though, that there is no such a palindrome.

You must write a program that gives the resulting palindrome and the number of iterations (additions) to compute the palindrome.

You might assume that all tests data on this problem:

- will have an answer
- will be computable with less than 1000 iterations (additions)
- will yield a palindrome that is not greater than 4,294,967,295.

Input

The first line will have a number N ($0 < N \leq 100$) with the number of test cases, the next N lines will have a number P to compute it's palindrome.

Output

For each of the N tests you will have to write a line with the following data : minimum number of iterations (additions) to get to the palindrome and the resulting palindrome itself separated by one space.

Sample Input

```
3
195
265
750
```

Sample Output

```
4 9339
5 45254
3 6666
```

Problem C: Mr. Ping and His Odd Hobby

Panda found out that the his dad (Mr. Ping) has a strange hobby! He loves playing with odd numbers. In the other day, he started writing, in each line, an odd number of odd numbers. It looked as follows:

1
3 5 7
9 11 13 15 17
19 21 23 25 27 29 31
. . .

On a certain line Mr. Ping wrote 55 odd numbers. Can you discover the sum of the last three numbers written in that line? Can you do this more generally for a given quantity of odd numbers? Given the number N of odd numbers in a certain line, your task is to determine the sum of the last three numbers of that line.

Input

The input is a sequence of lines, one odd number N $(1 < N < 1000000000)$ per line.

Output

For each input line write the sum of the last three odd numbers written by Mr. Ping in that line with N numbers. This sum is guaranteed to be less than 2^{63}.

Sample Input

```
3
5
7
```

Sample Output

```
15
45
87
```

Problem D: Po and The Damn Stairs of Jade Palace

“HOW MANY STAIRS DOES JADE PALACE HAVE???” said panda and the answer was 2^{31}. Lucky Master monkey that he can jump multiple stairs at once. Let’s call each jump a step. Poor Po! Po has invented a new game just for master monkey:

All stairs are numbered from 0 to 2^{31}. Master Po chooses a x and a y. Then master monkey should go from stair x to y but the length of a step must be non-negative and can be by one bigger than, equal to, or by one smaller than the length of the previous step. The length of the first and the last step must be 1.

You should write a program that gives the minimum number of steps from x to y

Input and Output

Input consists of a line containing n, the number of test cases. For each test case, a line follows with two integers: $0 \leq x \leq y < 2^{31}$. For each test case, print a line giving the minimum number of steps to get from x to y.

Sample Input

```
3
45  48
45  49
45  50
```

Sample Output

```
3
3
4
```

Problem E: A Panda Can't Fly!

Master Crane was one of the greatest Masters of Kung Fu in China. But he had to travel to another place for a month! It's a terrible disaster. As some of other valleys doesn't have any bridge or something for a walker to travel from the valley of peace. Dragon Warrior is now the protector of the valleys while master Crane is not at Jade Palace. Your duty is to write a program to find out traveling to what number of valleys are impossible for Po (Who can't fly)! Note that **Bridges are Unidirectional!!!**

A map is represented by n valleys where $1 \leq n \leq 100$, numbered consecutively $1, 2, ..., n$, and a series of one way bridges $p \rightarrow q$ which connect the pair of valleys p and q in one direction only.

A valley r is reachable from a valley p if there is an unidirectional $p \rightarrow r$, or if there exists some valley q for which q is reachable from p and r is reachable from q.

A valley r is inaccessible from a valley p if r is not reachable from p.

Input

The input data for this program consists of several Maps and starting valleys (As the Jade Palace Position).

For each map, ¡there is first one line containing a single integer n. This is the number of valleys in the map.

Following, there will be a group of lines, each containing a set of integers. The group is terminated by a line which contains only the integer 0. Each set represent a collection of bridges. The first integer in the set, i, is the starting valley (Valley of Peace), while the next group of integers, $j, ..., k$, define the series of bridges $i \rightarrow j, ..., i \rightarrow k$, and the last integer on the line is always 0. Each possible start valley i, $1 \leq i \leq n$ will appear once or not at all. Following each map definition, there will be a one line containing list of integers. The first integer on the line will specify how many integers follow. Each of the following integers represents a start valley (Valley of Peace) to be investigated by your program. The next maps then follows. If there are no more maps, the next line will contain only the integer 0.

Output

For each start valley to be investigated, your program should identify all the valleys which are inaccessible from the given start valley. Each list should appear on one line, beginning with the count of inaccessible valleys and followed by the inaccessible valleys numbers.

Sample Input

```
3
1 2 0
2 2 0
3 1 2 0
0
2 1 2
0
```

Sample Output

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

Problem F: Dragon Warrior's Tower

Perhaps you have heard of the legend of the Dragon Warrior's Tower. Nowadays many details of this tale have been forgotten. So now, in line with the educational and Shifuic-historical nature of this contest, we will tell you the whole story:

The people in the valley of peace had n types of blocks, and an unlimited supply of blocks of each type. Each type-i block was a rectangular solid with linear dimensions (x_i, y_i, z_i) . A block could be reoriented so that any two of its three dimensions determined the dimensions of the base and the other dimension was the height. They wanted to construct the tallest tower possible by stacking blocks. The problem was that, in building a tower, one block could only be placed on top of another block as long as the two base dimensions of the upper block were both strictly smaller than the corresponding base dimensions of the lower block. This meant, for example, that blocks oriented to have equal-sized bases couldn't be stacked.

Your job is to write a program that determines the height of the tallest tower the citizens of the valley of peace can build with a given set of blocks.

Input and Output

The input file will contain one or more test cases. The first line of each test case contains an integer n, representing the number of different blocks in the following data set. The maximum value for n is 30. Each of the next n lines contains three integers representing the values x_i , y_i and z_i .

Input is terminated by a value of zero (0) for n.

For each test case, print one line containing the case number (they are numbered sequentially starting from 1) and the height of the tallest possible tower in the format "Case case: maximum height = height"

Sample Input

```
1
10 20 30
2
6 8 10
5 5 5
7
1 1 1
2 2 2
3 3 3
4 4 4
5 5 5
6 6 6
7 7 7
5
31 41 59
26 53 58
97 93 23
84 62 64
33 83 27
0
```

Sample Output

```
Case 1: maximum height = 40
Case 2: maximum height = 21
Case 3: maximum height = 28
Case 4: maximum height = 342
```

Problem G: Po's Money

Po received money from Shifu this week and wants to spend it all buying books. But he does not read a book so fast, because he likes to enjoy every single word while he is reading. In this way, it takes him a week to finish a book. As Po receives money every two weeks, he decided to buy two books, then he can read them until receive more money. As he wishes to spend all the money, he should choose two books whose prices summed up are equal to the money that he has. It is a little bit difficult to find these books, so Po asks your help to find them.

Input

Each test case starts with $2 \leq N \leq 1000$, the number of available books. Next line will have N integers, representing the price of each book, a book costs less than 1000001. Then there is another line with an integer M, representing how much money Po has. There is a blank line after each test case. The input is terminated by end of file (EOF).

Output

For each test case you must print the message: 'Po should buy books whose prices are i and j.', where i and j are the prices of the books whose sum is equal do M and $i \leq j$. You can consider that is always possible to find a solution, if there are multiple solutions print the solution that minimizes the difference between the prices i and j . After each test case you must print a blank line.

Sample Input

```
2
40 40
80

5
10 2 6 8 4
10
```

Sample Output

```
Po should buy books whose prices are 40 and 40.

Po should buy books whose prices are 4 and 6.
```

Problem H: Everybody Knows Kung Fu Fighting!

Yes! Everybody knows kung fu fighting if we define it as some fists and kicks. But the truth beside it ...! OK, The Dragon Warrior is now gathering the secrets of the Universe in some scrolls. He wants to put them in a warehouse, So he designed new boxes to hold the scrolls inside. As the warehouse is so crowded he should put the boxes above each other.

So we have some boxes numbered 1 to N. The dimensions of all boxes are identical. Now we have to stack up some of the boxes, subject to the following constraints:

- Po cannot put more than one boxes directly upon a box;
- Boxes with lower numbers are not to be put upon one with a higher number;
- The weight and maximum load for each box are given. The total weight of all boxes upon a box should not exceed its maximum load.

Please write a program that finds the maximum number of boxes that can be stacked up according to the above constraints.

Input

The first line of each set of input is an integer N ($1 \leq N \leq 1000$). This is followed by N lines, each with two integers, both ≤ 3000, representing the weight and maximum load of each box respectively. Input ends with a case where $N = 0$.

Output

Each line of your output should give the number of boxes that can be stacked up.

Sample Input

```
5
19 15
7 13
5 7
6 8
1 2
0
```

Sample Output

```
4
```

Problem I: Mr. Ping's Farm

Mr. Ping has a farm where he farms materials of his delicious soup. But his farm is somehow irregular. His farm is divided to triangles and inside each triangle he plants a specific material. Now the problem is that he has lots of wooden bars to create a new triangle. he has n wooden bars in sizes $1, ..., n$. Know the question is how many distinct triangles he can make. Note that, two triangles will be considered different if they have at least 1 pair of arms with different length.

Input

The input for each case will have only a single positive integer n $(3 \leq n \leq 1000000)$. The end of input will be indicated by a case with $n < 3$. This case should not be processed.

Output

For each test case, print the number of distinct triangles you can make.

Sample Input

```
5
8
0
```

Sample Output

```
3
22
```

Problem J: Po's Money

Po received money from Shifu this week and wants to spend it all buying books. But he does not read a book so fast, because he likes to enjoy every single word while he is reading. In this way, it takes him a week to finish a book. As Po receives money every two weeks, he decided to buy two books, then he can read them until receive more money. As he wishes to spend all the money, he should choose two books whose prices summed up are equal to the money that he has. It is a little bit difficult to find these books, so Po asks your help to find them.

Input

Each test case starts with $2 \leq N \leq 100000$, the number of available books. Next line will have N integers, representing the price of each book, a book costs less than 1000001. Then there is another line with an integer M, representing how much money Po has. There is a blank line after each test case. The input is terminated by end of file (EOF).

Output

For each test case you must print the message: 'Po should buy books whose prices are i and j.', where i and j are the prices of the books whose sum is equal do M and $i \leq j$. You can consider that is always possible to find a solution, if there are multiple solutions print the solution that minimizes the difference between the prices i and j . After each test case you must print a blank line.

Sample Input

```
2
40 40
80

5
10 2 6 8 4
10
```

Sample Output

```
Po should buy books whose prices are 40 and 40.

Po should buy books whose prices are 4 and 6.
```

Problem K: Final Words in Oogway's Library

In Oogways library somewhere in Jade Palace there is a very strange scroll. Some ambigeous strings with a number. " You must find the nth palindrome from all sorted permutations the words and gather them all to reach the true power of Chi!! Oogway said!

But what is the sentence meaning? Let's ask Shifu! Maybe he knows the answer!

"We can permute the individual characters to make new strings. We can then order these strings into alphabetical order. " Shifu said!

For example the string 'abba' (The Spell of rising water from waterfall) gives rise to the following 6 distinct permutations in alphabetical order.

```
aabb 1
abab 2
abba 3
baab 4
baba 5
bbaa 6
```

Of these 6 permutations, only 2 are palindromes (A string that reads the same when read backwards). These are 'abba' and 'baab'. Given a string, you have to 2nd out the n^{th} palindrome in the sorted list of all permutations. For the above case 'abba' is the 1-st and 'baab' is the 2-nd palindrome.

Input

The first line of input gives the number of test cases. Each case contains a string, consisting of lowercase letters only, followed by a space separated positive integer n ($n < 2^{31}$). The length of the string will be at most 30.

Output

For each case, output the case number followed by the n-th palindrome, but if the total number of palindromes is less than n output 'XXX' without the quotes. Follow the sample for exact format.

Sample Input

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

Sample Output

```
Case 1: abba
Case 2: baab
Case 3: XXX
```

Problem L: Happy Ending in the Valley of Peace

In the valley of peace, people are obsessed with palindromes . There are N road junctions(also callled points) labeled 0 to $N-1$ and roads exist between every pair of points. Roads are onewayed and for the road connecting point i to point j ($i < j$) the direction to travel is i to j. Each road is labeled with a letter between 'A' to 'Z' . Lord Shen ,the traveler, wants to travel from point 0 to point $N-1$. However he wants to cover the longest palindromic path.

In the above arrangement the possible paths to take are:

```
ACCA
ABA
ACB
BCA
AD
BB
AA
C
```

The largest palindrome amongst these is ACCA, so Lord Shen will take this path. Given the above configuration, help him decide which path to take.

Input

The first line of input will contain an integer denoting the number of test cases $T \leq 25$. Each test case will be formatted as follows:-

- The first line of each test case contains an integer denoting $2 \leq N \leq 50$.
- The next N lines contain N characters each. Each character is a letter between 'A' to 'Z'. The j^{th} character in the i^{th} line denotes the label for the road between i to j and this will be equal to the i^{th} character in the j^{th} line.The i^{th} character of the i^{th} line will be * denoting no road exists.

Output

Output one line per case - The longest palindromic path available or "NO PALINDROMIC PATH" if none exists. Note that quotes are for clarity only. In case more than one longest path exists output the lexicographically smallest one.

Sample Input

```
2
5
*ABAC
A*CBD
BC*CB
ABC*A
CDBA*
5
*AXYZ
A*BQR
XB*BT
YQB*A
ZRTA*
```

Sample Output

```
ACCA
ABBA
```