A

Everyone probably knows the game *Zero or One* (in some regions in Brazil also known as *Two or One*), used to determine a winner among three or more players. For those unfamiliar, the game works as follows. Each player chooses a value between zero or one; prompted by a command (usually one of the contestants announces "Zero or... One!"), all participants show the value chosen using a hand: if the value chosen is one, the contestant shows a hand with an extended index finger; if the value chosen is zero, the contestant shows a hand with all fingers closed. The winner is the one who has chosen a value different from all others. If there is no player with a value different from all others (e.g. all players choose zero, or some players choose zero and some players choose one), there is no winner.

Alice, Bob and Clara are great friends and play Zerinho all the time: to determine who will buy popcorn during the movie session, who will enter the swimming pool first, etc.. They play so much that they decided make a plugin to play Zerinho on Facebook. But since the don't know how to program computers, they divided the tasks among friends who do know, including you.

Given the three values chosen by Alice, Bob and Clara, each value zero or one, write a program that determines if there is a winner, and in that case determines who is the winner.

Input

The input contains several test cases. Each test case contains a single line, with three integers A, B and C, indicating respectively the values chosen by Alice, Beto and Clara.

Output

For each test case your program must output a single line, containing a single character. If Alice is the winner the character must be '`A`', if Beto is the winner the character must be '`B`', if Clara is the winner the character must be '`C`', and if there is no winner the character must be '`*`' (asterisc).

Restrictions

- $A, B, C \in \{0, 1\}$

Sample Input

```
1 1 0
0 0 0
1 0 0
```

Sample Output

```
C
*
A
```

B

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 fiance 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 fiance 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 ($T \leq 100$), which indicates the number of test cases.

Each of the next T line contains three integers L, W and H ($1 \leq L, W, H \leq 50$) 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
```

Sample Output

```
Case 1: good
Case 2: bad
```

C

Mohammad has recently visited Switzerland. As he loves his friends very much, he decided to buy some chocolate for them, but as this fine chocolate is very expensive (You know Mohammad is a little BIT stingy!), he could only afford buying one chocolate, albeit a very big one (part of it can be seen in figure 1) for all of them as a souvenir. Now, he wants to give each of his friends exactly one part of this chocolate and as he believes all human beings are equal (!), he wants to split it into equal parts.

The chocolate is an $M \times N$ rectangle constructed from $M \times N$ unit-sized squares. You can assume that Mohammad has also $M \times N$ friends waiting to receive their piece of chocolate.

Figure 1. Mohammads chocolate

To split the chocolate, Mohammad can cut it in vertical or horizontal direction (through the lines that separate the squares). Then, he should do the same with each part separately until he reaches $M \times N$ unit size pieces of chocolate. Unfortunately, because he is a little lazy, he wants to use the minimum number of cuts required to accomplish this task.

Your goal is to tell him the minimum number of cuts needed to split all of the chocolate squares apart.

Input

The input consists of several test cases. In each line of input, there are two integers $1 \leq M \leq 300$, the number of rows in the chocolate and $1 \leq N \leq 300$, the number of columns in the chocolate. The input should be processed until end of file is encountered.

Output

For each line of input, your program should produce one line of output containing an integer indicating the minimum number of cuts needed to split the entire chocolate into unit size pieces.

Sample Input

```
2 2
1 1
1 5
```

Sample Output

```
3
0
4
```

D

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
```

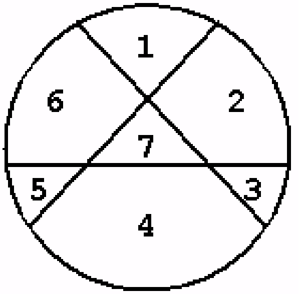
Sample Output

```
<
>
=
```

E

When someone calls Ivan lazy, he claims that it is his intelligence that helps him to be so. If his intelligence allows him to do something at less physical effort, why should he exert more? He also claims that he always uses his brain and tries to do some work at less effort; this is not his laziness, rather this is his intellectual smartness.

Once Ivan was asked to cut a pizza into seven pieces to distribute it among his friends. (Size of the pieces may not be the same. In fact, his piece will be larger than the others.) He thought a bit, and came to the conclusion that he can cut it into seven pieces by only three straight cuts through the pizza with a pizza knife. Accordingly, he cut the pizza in the following way (guess which one is Ivan's piece):



One of his friends, who never believed in Ivan's smartness, was startled at this intelligence. He thought, if Ivan can do it, why can't my computer? So he tried to do a similar (but not exactly as Ivan's, for Ivan will criticize him for stealing his idea) job with his computer. He wrote a program that took the number of straight cuts one makes through the pizza, and output a number representing the maximum number of pizza pieces it will produce.

Your job here is to write a similar program. It is ensured that Ivan's friend won't criticize you for doing the same job he did.

Input

The input file will contain a single integer N ($0 \leq N \leq 210000000$) in each line representing the number of straight line cuts one makes through the pizza. A negative number terminates the input.

Output

Output the maximum number of pizza pieces the given number of cuts can produce. Each line should contain only one output integer without any leading or trailing space.

Sample Input

```
5
10
-100
```

Sample Output

```
16
56
```

F

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!!
```

A square number is an integer number whose square root is also an integer. For example 1, 4, 81 are some square numbers. Given two numbers a and b you will have to find out how many square numbers are there between a and b (inclusive).

Input

The input file contains at most 201 lines of inputs. Each line contains two integers a and b ($0 < a \leq b \leq 100000$). Input is terminated by a line containing two zeroes. This line should not be processed.

Output

For each line of input produce one line of output. This line contains an integer which denotes how many square numbers are there between a and b (inclusive).

Sample Input

```
1 4
1 10
0 0
```

Sample Output

```
2
3
```

H

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 ($0 < n \leq 2000000$), 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 ($\sim$ 25 MB) so use faster IO.

Sample Input

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

Sample Output

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

Once upon a time, there is a special coco-cola store. If you return three empty bottles to the shop, you'll get a full bottle of coco-cola to drink. If you have n empty bottles right in your hand, how many full bottles of coco-cola can you drink?

Input

There will be at most 10 test cases, each containing a single line with an integer n ($1 \leq n \leq 100$). The input terminates with $n = 0$, which should not be processed.

Output

For each test case, print the number of full bottles of coco-cola that you can drink.

Spoiler

Let me tell you how to drink 5 full bottles with 10 empty bottles: get 3 full bottles with 9 empty bottles, drink them to get 3 empty bottles, and again get a full bottle from them. Now you have 2 empty bottles. *Borrow another empty bottle from the shop*, then get another full bottle. Drink it, and finally return this empty bottle to the shop!

Sample Input

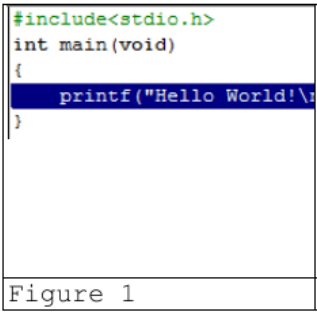
```
3
10
81
0
```

Sample Output

```
1
5
40
```

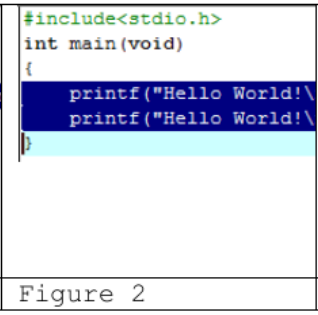
J

When you first made the computer to print the sentence “`Hello World!`”, you felt so happy, not knowing how complex and interesting the world of programming and algorithm will turn out to be. Then you did not know anything about loops, so to print 7 lines of “Hello World!”, you just had to copy and paste some lines. If you were intelligent enough, you could make a code that prints “Hello World!” 7 times, using just 3 paste commands. Note that we are not interested about the number of copy commands required. A simple program that prints “Hello World!” is shown in Figure 1. By copying the single print statement and pasting it we get a program that prints two “Hello World!” lines. Then copying these two print statements and pasting them, we get a program that prints four “Hello World!” lines. Then copying three of these four statements and pasting them we can get a program that prints seven “Hello World!” lines (Figure 4). So three pastes commands are needed in total and Of course you are not allowed to delete any line after pasting. Given the number of “Hello World!” lines you need to print, you will have to find out the minimum number of pastes required to make that program from the origin program shown in Figure 1.



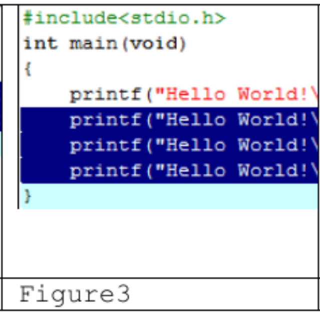
```
#include<stdio.h>
int main(void)
{
    printf("Hello World!\
}
```

Figure 1



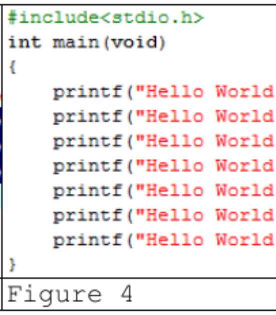
```
#include<stdio.h>
int main(void)
{
    printf("Hello World!\
    printf("Hello World!\
}
```

Figure 2



```
#include<stdio.h>
int main(void)
{
    printf("Hello World!\
    printf("Hello World!\
    printf("Hello World!\
    printf("Hello World!\
}
```

Figure3



```
#include<stdio.h>
int main(void)
{
    printf("Hello World
    printf("Hello World
    printf("Hello World
    printf("Hello World
    printf("Hello World
    printf("Hello World
    printf("Hello World
}
```

Figure 4

Input

The input file can contain up to 2000 lines of inputs. Each line contains an integer N ($0 < N < 10001$) that denotes the number of “Hello World!” lines are required to be printed.

Input is terminated by a line containing a negative integer.

Output

For each line of input except the last one, produce one line of output of the form ‘`Case` X`:` Y’ where X is the serial of output and Y denotes the minimum number of paste commands required to make a program that prints N lines of “Hello World!”.

Sample Input

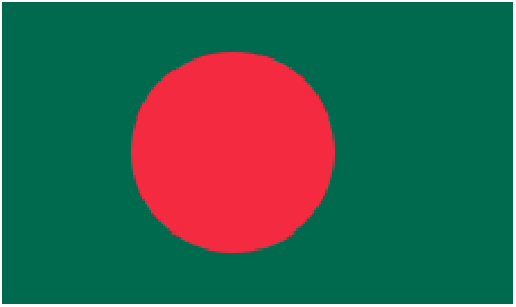
```
2
10
-1
```

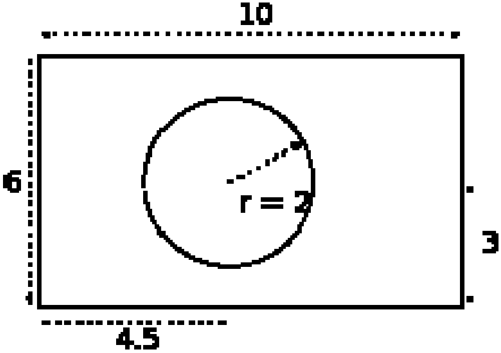
Sample Output

```
Case 1: 1
Case 2: 4
```

K

The national flag of Bangladesh is bottle green in color and rectangular in size with the **length** (L) to **width** ratio of 10:6. It bears a red circle on the background of green. It maintains the **length** (L) to **radius** ratio of 5:1 (If the **length** is 10 then **width** should be 6 and **radius** should be 2). The color in the background represents the greenery of Bangladesh while the red circle symbolizes the rising sun and the sacrifice of lives in our freedom fight.





Input

First line of input will contain the number of test cases, $T \leq 100$. Then there follows T lines, each containing a positive integer $L \leq 1000$, representing length of the flag.

Output

For each test case output is a line with two space separated real numbers containing exactly two digits after decimal point. Two numbers represent the area of red and green portion respectively.

Note: Pi is considered to be arccos(-1).

Sample Input

```
1
10
```

Sample Output

```
12.57 47.43
```

L

In Africa there is a very special species of bee. Every year, the female bees of such species give birth to one male bee, while the male bees give birth to one male bee and one female bee, and then they die!

Now scientists have accidentally found one “magical female bee” of such special species to the effect that she is immortal, but still able to give birth once a year as all the other female bees. The scientists would like to know how many bees there will be after N years. Please write a program that helps them find the number of male bees and the total number of all bees after N years.

Input

Each line of input contains an integer N (≥ 0). Input ends with a case where $N = -1$. (This case should NOT be processed.)

Output

Each line of output should have two numbers, the first one being the number of male bees after N years, and the second one being the total number of bees after N years. (The two numbers will not exceed 2^{32}.)

Sample Input

```
1
3
-1
```

Sample Output

```
1 2
4 7
```

M

Three families share a garden. They usually clean the garden together at the end of each week, but last week, family C was on holiday, so family A spent 5 hours, family B spent 4 hours and had everything done. After coming back, family C is willing to pay \$90 to the other two families. How much should family A get? You may assume both families were cleaning at the same speed.

\$90/(5+4)*5=\$50? No no no. Think hard. The correct answer is \$60. When you figured out why, answer the following question: If family A and B spent x and y hours respectively, and family C paid \$z, how much should family A get? It is guaranteed that both families should get non-negative integer dollars.

WARNING: Try to avoid floating-point numbers. If you really need to, be careful!

Input

The first line contains an integer T ($T \leq 100$), the number of test cases. Each test case contains three integers x, y, z ($1 \leq x, y \leq 10$, $1 \leq z \leq 1000$).

Output

For each test case, print an integer, representing the amount of dollars that family A should get.

Sample Input

```
2
5 4 90
8 4 123
```

Sample Output

```
60
123
```

N

The Antique Comedians of Malidinesia prefer comedies to tragedies. Unfortunately, most of the ancient plays are tragedies. Therefore the dramatic advisor of ACM has decided to transfigure some tragedies into comedies. Obviously, this work is very hard because the basic sense of the play must be kept intact, although all the things change to their opposites. For example the numbers: if any number appears in the tragedy, it must be converted to its reversed form before being accepted into the comedy play.

Reversed number is a number written in arabic numerals but the order of digits is reversed. The first digit becomes last and vice versa. For example, if the main hero had 1245 strawberries in the tragedy, he has 5421 of them now. Note that all the leading zeros are omitted. That means if the number ends with a zero, the zero is lost by reversing (e.g. 1200 gives 21). Also note that the reversed number never has any trailing zeros.

ACM needs to calculate with reversed numbers. Your task is to add two reversed numbers and output their reversed sum. Of course, the result is not unique because any particular number is a reversed form of several numbers (e.g. 21 could be 12, 120 or 1200 before reversing). Thus we must assume that no zeros were lost by reversing (e.g. assume that the original number was 12).

Input

The input consists of N cases. The first line of the input contains only positive integer N. Then follow the cases. Each case consists of exactly one line with two positive integers separated by space. These are the reversed numbers you are to add. Numbers will be at most 200 characters long.

Output

For each case, print exactly one line containing only one integer — the reversed sum of two reversed numbers. Omit any leading zeros in the output.

Sample Input

```
3
24 1
4358 754
305 794
```

Sample Output

```
34
1998
1
```

O

There is a village in Bangladesh, where brick game is very popular. Brick game is a team game. Each team consists of odd number of players. Number of players must be greater than 1 but cannot be greater than 10. Age of each player must be within 11 and 20. No two players can have the same age. There is a captain for each team. The communication gap between two players depends on their age difference, i.e. the communication gap is larger if the age difference is larger. Hence they select the captain of a team in such a way so that the number of players in the team who are younger than that captain is equal to the number of players who are older than that captain.

Ages of all members of the team are provided. You have to determine the age of the captain.

Input

Input starts with an integer T ($T \leq 100$), the number of test cases.

Each of the next T lines will start with an integer N ($1 < N < 11$), number of team members followed by N space separated integers representing ages of all of the members of a team. Each of these N integers will be between 11 and 20 (inclusive). Note that, ages will be given in strictly increasing order or strictly decreasing order. We will not mention which one is increasing and which one is decreasing, you have to be careful enough to handle both situations.

Output

For each test case, output one line in the format ‘`Case` x`:` a’ (quotes for clarity), where x is the case number and a is the age of the captain.

Sample Input

```
2
5 19 17 16 14 12
5 12 14 16 17 18
```

Sample Output

```
Case 1: 16
Case 2: 16
```

P

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:

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

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 ($0 \leq C, d \leq 100$), 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

```
2
100 0
0 100
```

Sample Output

```
Case 1: 100.00
Case 2: 55.56
```

Q

Problems in Computer Science are often classified as belonging to a certain class of problems (e.g., NP, Unsolvable, Recursive). In this problem you will be analyzing a property of an algorithm whose classification is not known for all possible inputs.

Consider the following algorithm:

```
1.  input $n$
2.  print $n$
3.  if $n = 1$ then STOP
4.      if $n$ is odd then $n \longleftarrow 3n + 1$
5.      else $n \longleftarrow n/2$
6.  GOTO 2
```

Given the input 22, the following sequence of numbers will be printed

$$22\ 11\ 34\ 17\ 52\ 26\ 13\ 40\ 20\ 10\ 5\ 16\ 8\ 4\ 2\ 1$$

It is conjectured that the algorithm above will terminate (when a 1 is printed) for any integral input value. Despite the simplicity of the algorithm, it is unknown whether this conjecture is true. It has been verified, however, for all integers n such that $0 < n < 1,000,000$ (and, in fact, for many more numbers than this.)

Given an input n, it is possible to determine the number of numbers printed before and including the 1 is printed. For a given n this is called the *cycle-length* of n. In the example above, the cycle length of 22 is 16.

For any two numbers i and j you are to determine the maximum cycle length over all numbers between and including both i and j.

Input

The input will consist of a series of pairs of integers i and j, one pair of integers per line. All integers will be less than 10,000 and greater than 0.

You should process all pairs of integers and for each pair determine the maximum cycle length over all integers between and including i and j.

You can assume that no operation overflows a 32-bit integer.

Output

For each pair of input integers i and j you should output i, j, and the maximum cycle length for integers between and including i and j. These three numbers should be separated by at least one space with all three numbers on one line and with one line of output for each line of input. The integers i and j must appear in the output in the same order in which they appeared in the input and should be followed by the maximum cycle length (on the same line).

Sample Input

```
1 10
100 200
201 210
900 1000
```

Sample Output

```
1 10 20
100 200 125
201 210 89
900 1000 174
```

R

A triangle is a geometric shape with three positive sides. However, any given three sides won't necessarily form a triangle. The three sides must form a closed region. Triangles are categorized depending on the values of the sides of a valid triangle. In this problem you are required to determine the type of a triangle.

Input

The first line of input will contain a positive integer $T < 20$, where T denotes the number of test cases. Each of the next T lines will contain three 32 bit signed integer.

Output

For each case of input there will be one line of output. It will be formatted as:

`Case` x`:` *triangle type*.

Where x denotes the case number being processed and *triangle type* is the type of the triangle. *triangle type* will be one of the following, depending on the values of the three sides:

- `Invalid` - The three sides can not form a triangle
- `Equilateral` - All three sides of valid triangle are equal
- `Isosceles` - Exactly two of the sides of a valid triangle are equal.
- `Scalene` - No pair of sides are equal in a valid triangle.

Sample Input

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

Sample Output

```
Case 1: Invalid
Case 2: Equilateral
Case 3: Isosceles
Case 4: Scalene
```

> *Labayk Allahuma Labayk. Labayk La shareeka laka Labayk. Innal hamda wannimata laka wal mulk. La shareeka Lak*
>
> (Here I am at your service, oh Lord, here I am - here I am. No partner do you have. Here I am. Truly, the praise and the favor are yours, and the dominion. No partner do you have.)

These are the words chanted by some two million people from world heading, as if pulled by a magnet, to one single spot on Earth. As has happened every year for 14 centuries, Muslim pilgrims gather in Makkah to perform rituals based on those conducted by the Prophet Muhammad during his last visit to the city.

Performing these rituals, known as the Hajj, is the fifth pillar of Islam and the most significant manifestation of Islamic faith and unity. Undertaking the Hajj at least once is a duty for Muslims who are physically and financially able to make the journey to Makkah. The emphasis on financial ability is meant to ensure that a Muslim takes care of his family first. The requirement that a Muslim be healthy and physically capable of undertaking the pilgrimage is intended to exempt those who cannot endure the rigors of extended travel.

The pilgrimage is the religious high point of a Muslim's life and an event that every Muslim dreams of undertaking. Umrah, the lesser pilgrimage, can be undertaken at any time of the year; Hajj, however, is performed during a five-day period from the ninth through the thirteenth of Dhu Al-Hijjah, the twelfth month of the Muslim lunar calendar.

It is generally presumed that the Hajj performed on Friday is called 'Hajj-e-Akbar' and it is a superior kind of Hajj as compared with the Hajj performed on other days of the week.

But, the correct meaning of the term, as explained by a large number of the commentators of the Holy Quran is that the Umrah, which can be performed at any time throughout the year, was generally called 'Hajj-e-Asghar' (the minor Hajj). In order to distinguish hajj from Umrah the former was named 'Hajj-e-Akbar' (the greater hajj). Therefore, each and every hajj is Hajj-e-Akbar, no matter whether it is performed on Friday or on any other day. The word 'Akbar' (greater) is used only to distinguish it from Umrah which is a minor Hajj.

Input

There will be several lines in the input terminated with a line containing a single '`*`'. This last line should not be processed. Each of the lines will contain either Hajj or Umrah.

Output

For each line of the input, output either '`Hajj-e-Akbar`' or '`Hajj-e-Asghar`' in separate lines without quotations. For exact format refer to the sample.

Sample Input

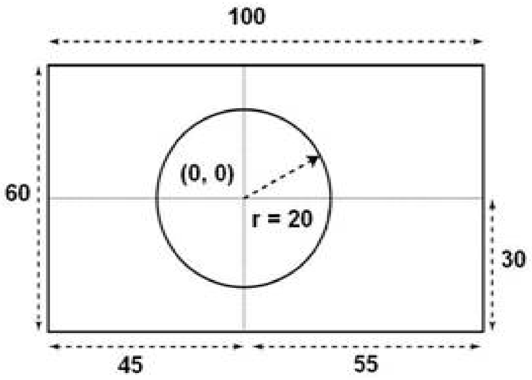
```
Hajj
Umrah
Hajj
Umrah
*
```

Sample Output

```
Case 1: Hajj-e-Akbar
Case 2: Hajj-e-Asghar
Case 3: Hajj-e-Akbar
Case 4: Hajj-e-Asghar
```

Teering is a little boy. He is trying to draw the national flag of Bangladesh. Being smart he knows he has to maintain the correct ratio and measurement while drawing the flag. You know the rules of drawing the national flag, don't you? If not, no worries, Teering is here to help you:

The national flag of Bangladesh consist of a green rectangle with a red circle inside it. The ratio of the length and width of the rectangle is **100 : 60** (i.e. if the length is 100 units then the width will be 60 units). The radius of the circle is **20%** of the length (i.e. if the length is 100 units then the radius of the circle will be 20 units). To get the center of the circle you need to draw a horizontal line dividing the **width in equal portion** and draw a vertical line dividing the **length in 45 : 55 ratio** (i.e. if the length of the rectangle is 100 then 45 units will be in left and 55 units will be on the right side of the line). The crossing of the two lines will be the center of the circle. Here is an illustrated picture for better understanding.



Now Teering has started to draw a flag. He has already drawn the circle of radius R centered at the **origin** in a 2D co-ordinate system. Now he needs to determine the corner of the rectangle so that he can join them to complete the flag. Can you help him?

Input

The first line of input will contain an integer T ($T < 101$) denoting the number of test cases. Each of the following T lines will contain an integer R ($R < 1001$) each denoting the radius of the circle.

Output

For each input output five lines. The first line will contain the case number. The following four lines will denote the **upper left**, **upper right**, **lower right** and **lower left** coordinates of the rectangle for the flag respectively. You have to print x coordinate and y coordinate separated by space in each line. You may assume that input is given in such that the corners will **always be in integer** coordinates. See sample input output for details.

Sample Input

```
2
20
100
```

Sample Output

```
Case 1:
-45 30
55 30
55 -30
-45 -30
Case 2:
-225 150
275 150
275 -150
-225 -150
```

U

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
```

Sample Output

```
101
10201
1101001
```

V

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 $3 (\text{mod} 2)$, or 1.

In this problem you have to calculate the parity of an integer $1 \leq I \leq 2147483647$.

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).
```

W

Trung is bored with his mathematics homeworks. He takes a piece of chalk and starts writing a sequence of consecutive integers starting with 1 to N ($1 < N < 10000$). After that, he counts the number of times each digit (0 to 9) appears in the sequence. For example, with $N = 13$, the sequence is:

$$12345678910111213$$

In this sequence, 0 appears once, 1 appears 6 times, 2 appears 2 times, 3 appears 3 times, and each digit from 4 to 9 appears once. After playing for a while, Trung gets bored again. He now wants to write a program to do this for him. Your task is to help him with writing this program.

Input

The input file consists of several data sets. The first line of the input file contains the number of data sets which is a positive integer and is not bigger than 20. The following lines describe the data sets.

For each test case, there is one single line containing the number N.

Output

For each test case, write sequentially in one line the number of digit $0, 1, \ldots 9$ separated by a space.

Sample Input

```
2
3
13
```

Sample Output

```
0 1 1 1 0 0 0 0 0 0
1 6 2 2 1 1 1 1 1 1
```

X

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?

Input

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

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

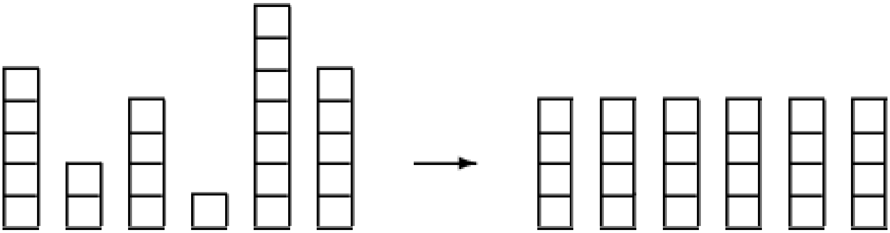
```
2
637
-120
```

Sample Output

```
1
3
```

Y

Little Bob likes playing with his box of bricks. He puts the bricks one upon another and builds stacks of different height. “Look, I’ve built a wall!”, he tells his older sister Alice. “Nah, you should make all stacks the same height. Then you would have a real wall.”, she retorts. After a little con- sideration, Bob sees that she is right. So he sets out to rearrange the bricks, one by one, such that all stacks are the same height afterwards. But since Bob is lazy he wants to do this with the minimum number of bricks moved. Can you help?



Input

The input consists of several data sets. Each set begins with a line containing the number n of stacks Bob has built. The next line contains n numbers, the heights h_i of the n stacks. You may assume $1 \leq n \leq 50$ and $1 \leq h_i \leq 100$.

The total number of bricks will be divisible by the number of stacks. Thus, it is always possible to rearrange the bricks such that all stacks have the same height.

The input is terminated by a set starting with $n = 0$. This set should not be processed.

Output

For each set, first print the number of the set, as shown in the sample output. Then print the line ‘`The minimum number of moves is` k`.`’, where k is the minimum number of bricks that have to be moved in order to make all the stacks the same height.

Output a blank line after each set.

Sample Input

```
6
5 2 4 1 7 5
0
```

Sample Output

```
Set #1
The minimum number of moves is 5.
```

A cow is grazing in the field. A rope in the field is tied with two pillars. The cow is kept tied with the rope with the help of a ring. So the cow can be considered to be tied with any point of the rope. Your job is to find the area of the field where the cow can reach and eat grass. If required assume that $\pi = 2 * \cos^{-1}(0)$ (Here angle is measured in radians). You can also assume that the thickness of the rope is zero, the cow is a point object and the radius of the ring and the thickness of the pillars are negligible. Please use double precision floating-point data type for floating-point calculations.

Input

First line of the input file contains an integer ($N \leq 100$), which indicates how many sets of inputs are there. Each of the next N lines contains two integers D ($0 \leq D \leq 1000$) and L ($D < L \leq 1500$). The first integer D denotes the distance in feet between the two pillars and the second integer L denotes the length of the rope in feet.

Output

Your program should produce N lines of output. Each line contains a single floating-point number, which has three digits after the decimal point. This floating-point number indicates the area of the field which the cow can reach and eat grass.

Sample Input

```
3
10 12
23 45
12 18
```

Sample Output

```
62.517
1366.999
189.670
```

Consider a polygon of equal sides inside a circle as shown in the figure below.

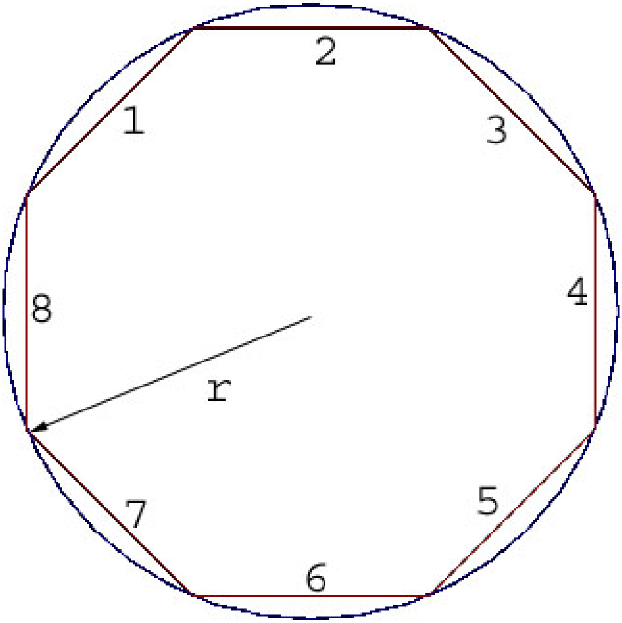


Figure: The regular polygon inside a circle

Given the radius of the circle and number of sides. You have to find the area of the polygon.

Input

In each line there will be two numbers indicating the radius r $(0 < r < 20000)$ and the number of sides of the polygon n $(2 < n < 20000)$ respectively. Input is terminated by EOF.

Output

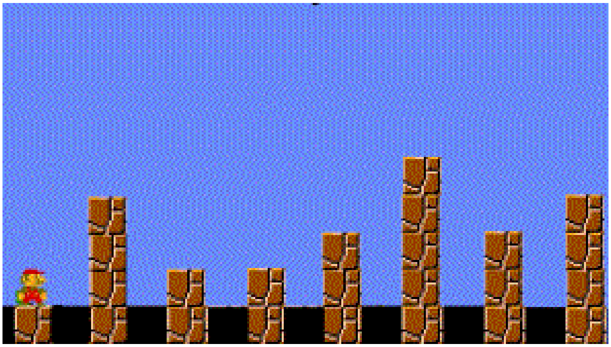
Output the area in each line. The number must be rounded to the third digit after the decimal point.

Sample Input

```
2 2000
10 3000
```

Sample Output

```
12.566
314.159
```



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

```
3
8
1 4 2 2 3 5 3 4
1
9
5
1 2 3 4 5
```

Sample Output

```
Case 1: 4 2
Case 2: 0 0
Case 3: 4 0
```

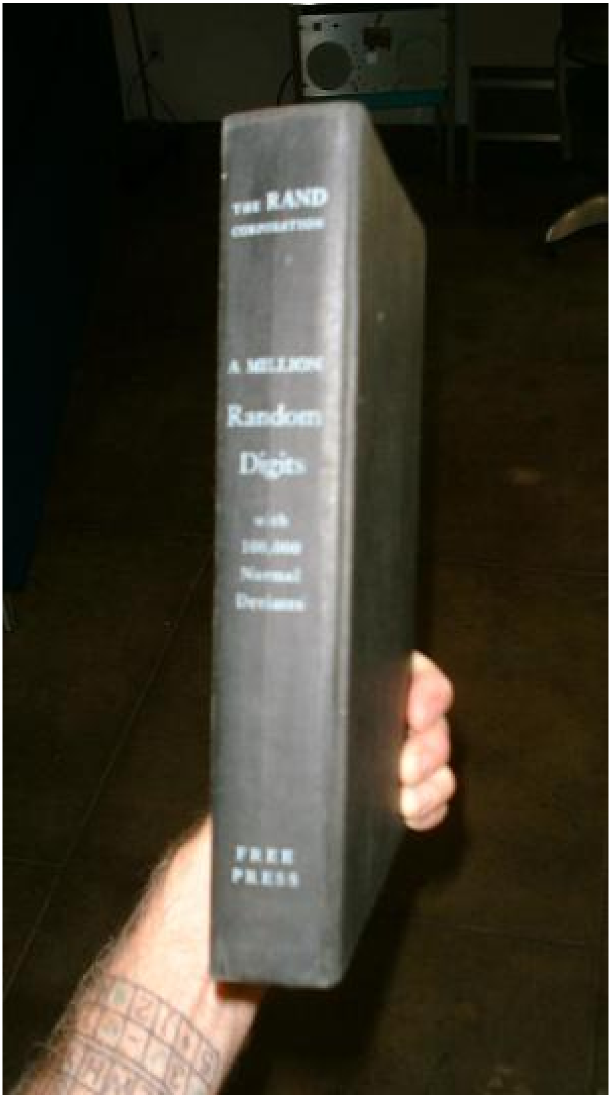
AC

For a positive integer n, let $f(n)$ denote the sum of the digits of n when represented in base 10. It is easy to see that the sequence of numbers $n, f(n), f(f(n)), f(f(f(n))), \ldots$ 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, $g(1234567892) = 2$.



Input

Each line of input contains a single positive integer n at most 2,000,000,000. Input is terminated by $n = 0$ which should not be processed.

Output

For each such integer, you are to output a single line containing $g(n)$.

Sample Input

```
2
11
47
1234567892
0
```

Sample Output

```
2
2
2
2
```

AD

The creators of a new programming language D++ have found out that whatever limit for SuperLongInt type they make, sometimes programmers need to operate even larger numbers. A limit of 1000 digits is so small... You have to find the sum of two numbers with maximal size of 1.000.000 digits.

Input

The first line of a input file is an integer N, then a blank line followed by N input blocks. The first line of an each input block contains a single number M $(1 \leq M \leq 1000000)$ the length of the integers (in order to make their lengths equal, some leading zeroes can be added). It is followed by these integers written in columns. That is, the next M lines contain two digits each, divided by a space. Each of the two given integers is not less than 1, and the length of their sum does not exceed M.

There is a blank line between input blocks.

Output

Each output block should contain exactly M digits in a single line representing the sum of these two integers.

There is a blank line between output blocks.

Sample Input

```
2

4
0 4
4 2
6 8
3 7

3
3 0
7 9
2 8
```

Sample Output

```
4750

470
```

AE

Given the length of three medians of a triangle you will have to find out the area of the triangle. Unless you are weak in geometry you should know that median of a triangle is formed by connecting any vertex of a triangle and the mid-point of its opposite edge. So a triangle has three medians.

Input

The input file contains 1000 lines of input. Each line contains three numbers which denote the length of the medians of a triangle. All the values in the input will be less than 100. Input is terminated by end of file.

Output

For each line of input you should produce one line of output. This line should contain the area of the triangle for the corresponding input. If it is not possible to form a triangle with the given medians, the area of the triangle should be set as ‘`-1`. The areas should be rounded up to three digits after the decimal point. Consider that a triangle must have a STRICTLY positive area.

Sample Input

```
3 3 3
3 3 3
```

Sample Output

```
5.196
5.196
```

AF

n people board an airplane with n seats. The first passenger has lost his boarding pass, so he sits in a random seat. Each subsequent passenger sits in his own seat if it's available or takes a random unoccupied seat if it's not.

What's the probability that the nth passenger finds his seat occupied?

Input

The input file contains several test cases. Each test case is described with one integer n on a single line $(2 \leq n \leq 1000)$.

The last line contains a single '`0`' and should not be processed.

Output

For each test case, output the probability that the n-th passenger finds his seat occupied on a single line.

If the probability is 0, output '`0/1`'. Otherwise, the probability should be expressed as an irreducible fraction a/b, where a and b are positive integers and a and b be are relatively prime. Do not print any spaces between the numbers or the division sign.

Sample Input

```
2
0
```

Sample Output

```
1/2
```

Prop hunt is a game modification (or simply a mod), similar to the famous hide and seek, where two teams play against each other. One team are the props, they are players that can choose any object of the scenario to disguise themselves as, so they could be a table, a book, a barrel, a paper, whatever they want. They must be completely calm or the other team, the hunters, will kill them. As a hunter, you must be careful: if you shoot something that is not a disguised player, you start losing life. At the end, if any prop survives they will win, otherwise the hunters will win.

Your mission is to determine who is going to be the winner of the game, but thankfully, with a couple of simplifications. You will get three numbers: P — the quantity of players in the props team, H — the number of hunters and O — the objects to disguise as. Assume that all the hunters are bots (that is, controlled by the computer) and they're pretty bad: the hunters will shoot everything in the scenario, starting by the less suspicious objects and ending with the disguised players. If every hunter has 1 point of life, what team will win?

Input

The input consists of several test cases, each one on a single line that contains 3 integers P $(1 \leq P \leq 5)$, H $(1 \leq H \leq 5)$ and O $(1 \leq P \leq O \leq 10)$.

Output

Print on a single line the text '`Props win!`' if the props survive, otherwise print '`Hunters win!`'.

Explanation: On the first case, there are 2 props and 2 hunters, but 3 objects to disguise as. The first hunter dies trying to kill the object that is not a player, but the second hunter shoots to the other objects that are the disguised players.

In the second case, the first hunter shoots a non-player object, and the second one does the same, so the props survive.

Sample Input

```
2 2 3
2 2 4
```

Sample Output

```
Hunters win!
Props win!
```

AH

Write a program that will reverse the letters in each of a sequence of words while preserving the order of the words themselves.

Input

The input file will consist of several lines of several words. Words are contiguous stretches of printable characters delimited by white space.

Output

The output will consist of the same lines and words as the input file. However, the letters within each word must be reversed.

Sample Input

```
I love you.
You love me.
We're a happy family.
```

Sample Output

```
I evol .uoy
uoY evol .em
er'eW a yppah .ylimaf
```

AI

Banglawash! It's a popular term used by the fans of Bangladesh cricket team when their team achieved a rare but much expected clean sweep over their opponent. In this November, 2013, the touring New Zealand cricket team is once again *Banglawased* in the 3-match ODI series.

In cricket, the term *whitewash* is used when one team wins all the matches played in a particular series; obviously abandoned match are not counted. Apart from this year's achievement, Bangladesh defeated New Zealand 4 − 0 to win a 5-match ODI series (one game was abandoned) in October 2010. New Zealand was touring Bangladesh on both the occasions. These two series were labeled as *Banglawash* as the language of Bangladesh is Bangla and 'Bangla' often used among locals for the things made in Bangladesh.

Some of the recent examples of *whitewash* include the following.

England's 4-0 defeat of India to win the Pataudi Trophy during India's tour of England in 2011.

Australia's 4-0 defeat of India to win the 2011-12 Border-Gavaskar Trophy.

India's 4-0 defeat of Australia to win the 2012-13 Border-Gavaskar Trophy.

West Indies' consecutive 5 − 0 defeats of England in 1984 and 1985-86. These two results are also commonly labeled as *blackwashes* because of the dark skin of the West Indies players.

For the cricket enthusiastic fan of the Tigers, here is a list of eight *Banglawash*.

Opponent	**Year**	**Result**
Kenya	2006	4 − 0
Kenya	2006	3 − 0
Zimbabwe	2006	5 − 0
Scotland	2006	2 − 0
Ireland	2008	3 − 0
West Indies	2009	3 − 0
New Zealand	2010	4 − 0
New Zealand	2013	3 − 0

Now, Bangladesh is playing against the Rest of the World team, which is formed by taking players from World Wide Web (WWW) of cricketers. Hence, the team name is WWW.

Input

First line of the input contains one positive integer T the number of test cases. The first line of each test case contains one positive integer N. N denotes how many match have been played in a single series. In the next line, there will be N uppercase letters. These letters will be either '`B`' or '`W`' or '`T`' or '`A`'. Here, '`B`' means the match was won by Bangladesh, '`W`' means the match was won by WWW, '`T`' means the match was a tie and '`A`' means the match was abandoned.

Constraints

- $T < 101$
- $N < 11$

Output

For each test case output the final outcome of the series. The outcome can be any one of the following.

'`BANGLAWASH`' – If Bangladesh won all the matches played (excluding the abandoned matches) in the series

'`WHITEWASH`' – If WWW won all the matches played (excluding the abandoned matches) in the series

'`ABANDONED`' – If all the matches of the series is abandoned

'`DRAW`' – If Bangladesh and WWW won equal number of matches in the series when the series is not abandoned

'`BANGLADESH`' – If Bangladesh won more matches then WWW in the series but failed to achieve a Banglawash

'`WWW`' – If WWW won more matches then Bangladesh in the series but failed to achieve a Whitewash

For '`BANGLADESH`' or '`WWW`' you have to show the number of match won by each country. And for '`DRAW`' you have to show how many wins and how many ties by each country.

For exact format see the sample output.

Sample Input

```
6
3
BBB
3
WWW
3
BWB
4
BWWT
3
BTW
2
AA
```

Sample Output

```
Case 1: BANGLAWASH Case 2: WHITEWASH Case 3: BANGLADESH 2 - 1 Case 4: WWW 2 - 1
Case 5: DRAW 1 1 Case 6: ABANDONED
```

The inhabitants of Nlogonia are very superstitious. One of their beliefs is that street house numbers that have a repeated digit bring bad luck for the residents. Therefore, they would never live in a house which has a street number like 838 or 1004.

The Queen of Nlogonia ordered a new seaside avenue to be built, and wants to assign to the new houses only numbers without repeated digits, to avoid discomfort among her subjects. You have been appointed by Her Majesty to write a program that, given two integers N and M, determines the maximum number of houses that can be assigned street numbers between N and M, inclusive, that do not have repeated digits.

Input

Each test case is described using one line. The line contains two integers N and M, as described above ($1 \leq N \leq M \leq 5000$).

Output

For each test case output a line with an integer representing the number of street house numbers between N and M, inclusive, with no repeated digits.

Sample Input

```
87 104
989 1022
22 25
1234 1234
```

Sample Output

```
14
0
3
1
```

AK

On the dining table we have a packet of soya milk. We can look at the packet as a rectangular box with height h, length l and width w. We tilt it against a base edge, as shown in the picture. Soya milk then comes out through the opening in the top face. This process stops when the milk level is no longer higher than the opening.

Write a program that computes the volume of soya milk remaining in the packet.

Note that the problem description may not perfectly match the real-life situation, but to solve this problem you must follow our assumptions.

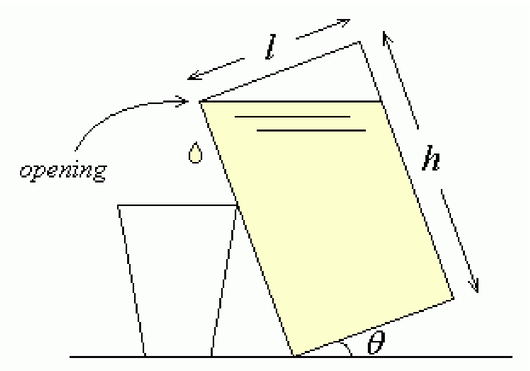
Input

Each input file contains multiple test cases, and each test case starts on a new line.

Each case consists of four positive integers l, w, h (in centimetres) and θ (in degrees), all strictly less than 90.

Output

For each test case, output a line containing the volume of soya milk inside the packet in millilitres, correct to 3 decimal places.



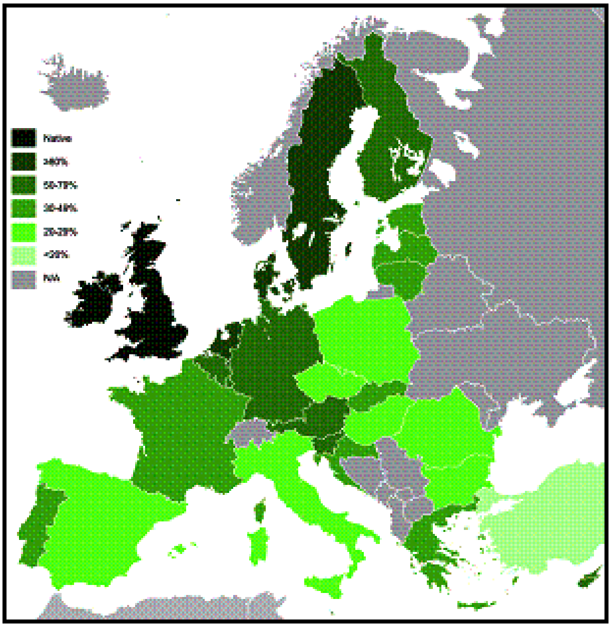
Sample Input

```
9 6 19 20
```

Sample Output

```
937.555 mL
```

AL



English, Spanish, German, French, Italian and Russian are the 6 most prominent languages in the countries of European Union. Figure on the left shows intensity of English speaking people in different European countries. All of these languages have different words to represent the English word "HELLO". For example in Spanish the word equivalent to "HELLO" is "HOLA". In German, French, Italian and Russian language the word that means (or similar to) "HELLO" is "HALLO", "BONJOUR", "CIAO" and "ZDRAVSTVUJTE" respectively. In this problem your task is pretty simple. You will be given one of the six words mentioned above or any other word and you will have to try and detect the language it is from.

Input

Input file contains around 2000 lines of inputs. Each line contains a string S. You can assume that all the letters of the string are uppercase English letters and the maximum length of the string is 14. Input is terminated by a line containing a single '#' character (without the quote). This line should not be processed.

Output

For each line of input except the last one produce one line of output. This line contains the serial of output followed by a language name. If the input string is 'HELLO' or 'HOLA' or 'HALLO' or 'BONJOUR' or 'CIAO' or 'ZDRAVSTVUJTE' then you should report the language it belongs to. If the input string is something other than these 6 strings print the string 'UNKNOWN' (without the quotes) instead. All characters in the output strings are uppercase as well. Look at the output for sample input for formatting details.

Sample Input

```
HELLO
HOLA
HALLO
BONJOUR
CIAO
ZDRAVSTVUJTE
#
```

Sample Output

```
Case 1: ENGLISH
Case 2: SPANISH
Case 3: GERMAN
Case 4: FRENCH
Case 5: ITALIAN
Case 6: RUSSIAN
```

AM

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

Input

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 ($1 \leq N \leq 1000$). 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.

Output

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
```

Sample Output

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

AN

You are in a car and going at the speed of u m/s. Your acceleration a is constant. After a particular time t, your speed is v m/s and your displacement is s. Now you are given some (not all of them) values for the given variables. And you have to find the missing parameters.

Input

The input file may contain multiple test cases. Each test case can be one of the

1 u v t
2 u v a
3 u a s
4 v a s

Input will be terminated by a single '0'.

Output

For each case of input you have to print one line containing the case number and

If 1 u v t are given then print s and a
If 2 u v a are given then print s and t
If 3 u a s are given then print v and t
If 4 v a s are given then print u and t

Check the samples for more details. You can assume that the given cases will not evaluate to an invalid situation. Use double for all calculations and output all floating point numbers to three decimal places.

Sample Input

```
1 10 5 2.0
1 5 10.0 2
2 10 11 2
3 5 1 6
4 5.0 -1 6
0
```

Sample Output

```
Case 1: 15.000 -2.500
Case 2: 15.000 2.500
Case 3: 5.250 0.500
Case 4: 6.083 1.083
Case 5: 6.083 1.083
```

AO

In my country, streets dont have names, each of them are just given a number as name. These numbers are supposed to be unique but that is not always the case. The local government allocates some integers to name the roads and in many case the number of integers allocated is less that the total number of roads. In that case to make road names unique some single character suffixes are used. So roads are named as 1, 2, 3, 1A, 2B, 3C, etc. Of course the number of suffixes is also always limited to 26 (A, B, . . . , Z). For example if there are 4 roads and 2 different integers are allocated for naming then some possible assignments of names can be:

1, 2, 1A, 2B
1, 2, 1A, 2C
3, 4, 3A, 4A
1, 2, 1B, 1C

Given the number of roads (R) and the numbers of integers allocated for naming (N), your job is to determine minimum how many different suffixes will be required (of all possible namings) to name the streets assuming that no two streets can have same names.

Input

The input file can contain up to 10002 lines of inputs. Each line contains two integers R and N $(0 < N, R < 10001)$. Here R is the total number of streets to be named and N denotes number integers allocated for naming.

Output

For each line of input produce one line of output. This line contains the serial of output followed by an integer D which denotes the minimum number of suffixes required to name the streets. If it is not possible to name all the streets print '`impossible`' instead (without the quotes).

Sample Input

```
8 5
100 2
0 0
```

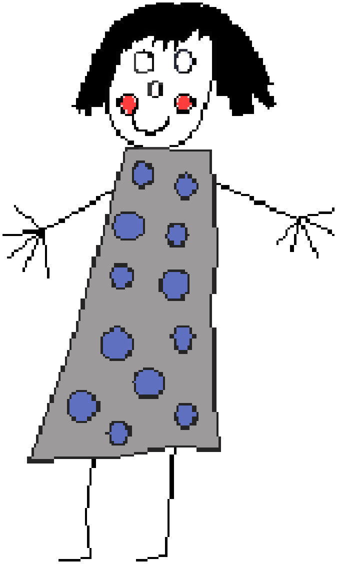
Sample Output

```
Case 1: 1
Case 2: impossible
```

AP

Joana loves playing with odd numbers. In the other day, she 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 Joana 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 Joana 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
```