SOME OF EASY UVA PROBLEMS

Set	: 02
Problem Type	: Ad Hoc
Problem Range	: 10000 – 10999
Total Problem	: 72

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

INDEX OF PROBLEMS

37. 10407 - Simple division
38. 10409 - Die Game
39. 10420 - List of Conquests
40. 10424 - Love Calculator
41. 10432 - Polygon Inside A Circle
42. 10451 - Ancient Village Sports
43. 10461 - Difference
44. 10473 - Simple Base Conversion
45. 10490 - Mr. Azad and his Son!!!!!
46. 10499 - The Land of Justice
47. 10515 - Powers Et Al.
48. 10522 - Height to Area
49. 10533 - Digit Primes
50. 10573 - Geometry Paradox
51. 10591 - Happy Number
52. 10611 - The Playboy Chimp
53. 10678 - The Grazing Cow
54. 10679 - I Love Strings!!
55. 10683 - The decadary watch
56. 10689 - Yet another Number Sequence
57. 10696 - f91
58. 10699 - Count the factors
59. 10700 - Camel trading
60. 10703 - Free spots
61. 10784 - Diagonal
62. 10789 - Prime Frequency
63. 10790 - How Many Points of Intersection?
64. 10812 - Beat the Spread!
65. 10815 - Andy's First Dictionary
66. 10921 - Find the Telephone
67. 10922 - 2 the 9s
68. 10924 - Prime Words
69. 10929 - You can say 11
70. 10931 - Parity
71. 10970 - Big Chocolate
72. 10991 - Region

1. 10006 - Carmichael Numbers

Time limit: 3.000 seconds

Carmichael Numbers

An important topic nowadays in computer science is cryptography. Some people even think that cryptography is the only important field in computer science, and that life would not matter at all without cryptography. Alvaro is one of such persons, and is designing a set of cryptographic procedures for cooking paella. Some of the cryptographic algorithms he is implementing make use of big prime numbers. However, checking if a big number is prime is not so easy. An exhaustive approach can require the division of the number by all the prime numbers smaller or equal than its square root. For big numbers, the amount of time and storage needed for such operations would certainly ruin the paella.

However, some probabilistic tests exist that offer high confidence at low cost. One of them is the Fermat test.

Let *a* be a random number between 2 and *n* - 1 (being *n* the number whose primality we are testing). Then, *n* is probably prime if the following equation holds:

$$a^n \bmod n = a$$

If a number passes the Fermat test several times then it is prime with a high probability.

Unfortunately, there are bad news. Some numbers that are not prime still pass the Fermat test with every number smaller than themselves. These numbers are called Carmichael numbers.

In this problem you are asked to write a program to test if a given number is a Carmichael number. Hopefully, the teams that fulfill the task will one day be able to taste a delicious portion of encrypted paella. As a side note, we need to mention that, according to Alvaro, the main advantage of encrypted paella over conventional paella is that nobody but you knows what you are eating.

Input

The input will consist of a series of lines, each containing a small positive number *n* (2 < *n* < 65000). A number *n* = 0 will mark the end of the input, and must not be processed.

Output

For each number in the input, you have to print if it is a Carmichael number or not, as shown in the sample output.

Sample Input

```
1729
17
561
1109
431
0
```

Sample Output

```
The number 1729 is a Carmichael number.
17 is normal.
The number 561 is a Carmichael number.
1109 is normal.
431 is normal.
```

2. 10008 - What's Cryptanalysis?

Time limit: 3.000 seconds

What's Cryptanalysis?

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

Input

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

Output

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

Sample Input

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

Sample Output

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

3. 10013 - Super long sums

Time limit: 3.000 seconds

Super long sums

The Problem

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.

The 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<=M<=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.

The 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
```

4. 10019 - Funny Encryption Method

Time limit: 3.000 seconds

Funny Encryption Method

The Problem

History :

A student from ITESM Campus Monterrey plays with a new encryption method for numbers. These method consist of the following steps:

Steps : Example
1) Read the number N to encrypt M = 265
2) Interpret N as a decimal number X1= 265 (decimal)
3) Convert the decimal interpretation of N to its binary representation X1= 100001001 (binary)
4) Let b1 be equal to the number of 1's in this binary representation B1= 3
5) Interpret N as a Hexadecimal number X2 = 265 (hexadecimal)
6) Convert the hexadecimal interpretation of N to its binary representation X2 = 1001100101
7) Let b2 be equal to the number of 1's in the last binary representation B2 = 5
8) The encryption is the result of M xor (b1*b2) M xor (3*5) = 262

This student failed Computational Organization, that's why this student asked the judges of ITESM Campus Monterrey internal ACM programming Contest to ask for the numbers of 1's bits of this two representations so that he can continue playing.

Task :
You have to write a program that read a Number and give as output the number b1 and b2

The Input

The first line will contain a number N which is the number of cases that you have to process. Each of the following N Lines (0<N<=1000) will contain the number M (0<M<=9999, in decimal representation) which is the number the student wants to encrypt.

The Output

You will have to output N lines, each containing the number b1 and b2 in that order, separated by one space corresponding to that lines number to crypt

Sample Input

```
3
265
111
1234
```

Sample Output

```
3 5
6 3
5 5
```

5. 10023 - Square root

Time limit: 3.000 seconds

Square root

The Problem

You are to determinate X by given Y, from expression $X=\sqrt{Y}$

The Input

The first line is the number of test cases, followed by a blank line.

Each test case of the input contains a positive integer Y ($1<=Y<=10^{1000}$), with no blanks or leading zeroes in it.
It is guaranteed, that for given Y, X will be always an integer.

Each test case will be separated by a single line.

The Output

For each test case, your program should print X in the same format as Y was given in input.

Print a blank line between the outputs for two consecutive test cases.

Sample Input

```
1

7206604678144
```

Sample Output

```
2684512
```

6. 10060 - A hole to catch a man

Time limit: 3.000 seconds

How can a manhole be a hole if it is covered? Perhaps, to prove a manhole a hole, most of the manholes of Dhaka are uncovered. So now manhole means *a hole to catch a man*. Anyway, the new Mayor of Dhaka does not like this definition and he has recently been highly acclaimed by general people for ordering corresponding department to cover all the manholes of the city within a month.

Manhole Cover Manufacturing Corporation (*MCMC*) somehow managed to get the order. (Yes, this is a big deal, since a lot of manhole covers are to be made). MCMC makes the cover using steel, and they import polygonal steel sheets of different shapes and thickness from abroad. Then they melt the sheets to make the circular manhole covers, which also differ in size and thickness.

MCMC needs a program which, given dimensions of a number of steel sheets, will calculate how many manhole cover can be made from these sheets. You are to help them by writing the program.

Input

The input file consists of several data blocks.

Each data block starts with an integer N, the number of polygonal steel sheets.i'th line of the next N lines starts with thickness of the i'th sheet followed by co-ordinates of the polygons' corner points in some order (clockwise or anti-clockwise). Each line consists of a series of real numbers in following format:

$$T_i\ X_0\ Y_0\ X_1\ Y_1\ X_2\ Y_2\ \ldots\ \ldots\ X_n\ Y_n\ X_0\ Y_0$$

Where T_i is the thickness of the sheet, and $X_i\ Y_i$ are the coordinates of corner points. The line ends with co-ordinate of the first point. Last line of each data block will have two real numbers, R and T, radius and thickness of the manhole cover respectively.

Input file ends with a data block with $N = 0$.

Output

For each data block, print the number of manhole cover in separate line.

Sample Input:

```
2
2 0 0 0 10 5 15 12 10 10 0 0 0
5 0 0 5 100 100 0 0 0
5 3
1
2 0 0 10 0 10 10 0 10 0 0
5 2
0
```

Sample Output:

```
107
1
```

7. 10062 - Tell me the frequencies!

Time limit: 3.000 seconds

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

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

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

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

Sample Input:
AAABBC
122333

Sample Output:
67 1
66 2
65 3

49 1
50 2
51 3

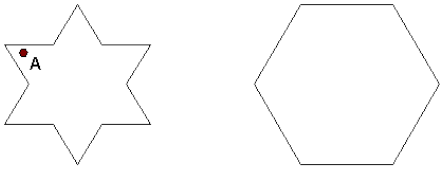
Shahriar Manzoor
16-12-2000

8. 10078 - The Art Gallery

Time limit: 3.000 seconds

Century Arts has hundreds of art galleries scattered all around the country and you are hired to write a program that determines whether any of the galleries has a *critical point*. The galleries are polygonal in shape and a *critical point* is a point inside that polygon from where the entire gallery is not visible.

For example, in gallery 1 (drawn below) point *A* is a critical point, but gallery 2 has no critical point at all.



Gallery 1 Gallery 2

The *Century Arts* authorities will provide you with the shape of a gallery as a sequence of (x, y) co-ordinates (determined using a suitable origin) of the consecutive corner points of that gallery.

Input

The input file consists of several data blocks. Each data block describes one gallery.

The first line of a data block contains an integer N ($3 \le N \le 50$) indicating the number of corner points of the gallery. Each of the next N lines contains two integers giving the (x, y) co-ordinates of a corner point where $0 \le x, y \le 1000$. Starting from the first point given in the input the corner points occur in the same order on the boundary of the gallery as they appear in the input. No three consecutive points are co-linear.

The input file terminates with a value of 0 for N.

Output

For each gallery in the input output the word "*Yes*" if the gallery contains a critical point, otherwise output the word "*No*". Each output must be on a separate line.

Sample Input

```
4
0 0
3 0
3 3
0 3
4
0 0
3 0
1 1
0 3
0
```

Sample Output

```
No
Yes
```

9. 10093 - An Easy Problem!

Time limit: 3.000 seconds

Problem I
An Easy Problem!
Input: Standard Input
Output: Standard Output

Have you heard the fact “The base of every normal number system is 10” ? Of course, I am not talking about number systems like Stern Brockot Number System. This problem has nothing to do with this fact but may have some similarity.

You will be given an N based integer number R and you are given the guaranty that R is divisible by (N-1). You will have to print the smallest possible value for N. The range for N is 2<=N<=62 and the digit symbols for 62 based number is (0..9 and A..Z and a..z). Similarly, the digit symbols for 61 based number system is (0..9 and A..Z and a..y) and so on.

Input
Each line in the input file will contain an integer (as defined in mathematics) number of any integer base (2..62). You will have to determine what is the smallest possible base of that number for the given conditions. No invalid number will be given as input.

Output
If number with such condition is not possible output the line “such number is impossible!” For each line of input there will be only a single line of output. The output will always be in decimal number system.

Sample Input
```
3
5
A
```

Sample Output
```
4
6
11
```

Shahriar Manzoor

"A machine can do the work of fifty ordinary men, but no machine can do the work of one extra-ordinary man."

10. 10101 - Bangla Numbers

Time limit: 3.000 seconds

Bangla numbers normally use 'kuti' (10000000), 'lakh' (100000), 'hajar' (1000), 'shata' (100) while expanding and converting to text. You are going to write a program to convert a given number to text with them.

Input
The input file may contain several test cases. Each case will contain a non-negative number <= 999999999999999.

Output
For each case of input, you have to output a line starting with the case number with four digits adjustment followed by the converted text.

Sample Input
```
23764
45897458973958
```

Sample Output
```
   1. 23 hajar 7 shata 64
   2. 45 lakh 89 hajar 7 shata 45 kuti 89 lakh 73 hajar 9 shata 58
```

11. 10104 - Euclid Problem

Time limit: 3.000 seconds

The Problem

From Euclid it is known that for any positive integers A and B there exist such integers X and Y that $AX+BY=D$, where D is the greatest common divisor of A and B. The problem is to find for given A and B corresponding X, Y and D.

The Input

The input will consist of a set of lines with the integer numbers A and B, separated with space ($A,B<1000000001$).

The Output

For each input line the output line should consist of three integers X, Y and D,separated with space. If there are several such X and Y, you should output that pair for which $|X|+|Y|$ is the minimal (primarily) and $X<=Y$ (secondarily).

Sample Input
```
4 6
17 17
```

Sample Output
```
-1 1 2
0 1 17
```

12. 10107 - What is the Median?

Time limit: 3.000 seconds

What is the Median?

The Problem

Median plays an important role in the world of statistics. By definition, it is a value which divides an array into two equal parts. In this problem you are to determine the current median of some long integers.

Suppose, we have five numbers {1,3,6,2,7}. In this case, 3 is the median as it has exactly two numbers on its each side. {1,2} and {6,7}.

If there are even number of values like {1,3,6,2,7,8}, only one value cannot split this array into equal two parts, so we consider the average of the middle values {3,6}. Thus, the median will be (3+6)/2 = 4.5. In this problem, you have to print only the integer part, not the fractional. As a result, according to this problem, the median will be 4!

Input

The input file consists of series of integers X (0 <= X < 2^31) and total number of integers N is less than 10000. The numbers may have leading or trailing spaces.

Output

For each input print the current value of the median.

Sample Input

```
1
3
4
60
70
50
2
```

Sample Output

```
1
2
3
3
4
27
4
```

Sadi Khan

2001-04-01

13. 10110 - Light, more light

Time limit: 3.000 seconds

Light, more light

The Problem

There is man named "mabu" for switching on-off light in our University. He switches on-off the lights in a corridor. Every bulb has its own toggle switch. That is, if it is pressed then the bulb turns on. Another press will turn it off. To save power consumption (or may be he is mad or something else) he does a peculiar thing. If in a corridor there is `n' bulbs, he walks along the corridor back and forth `n' times and in i'th walk he toggles only the switches whose serial is divisable by i. He does not press any switch when coming back to his initial position. A i'th walk is defined as going down the corridor (while doing the peculiar thing) and coming back again.

Now you have to determine what is the final condition of the last bulb. Is it on or off?

The Input

The input will be an integer indicating the n'th bulb in a corridor. Which is less then or equals 2^32-1. A zero indicates the end of input. You should not process this input.

The Output

Output "yes" if the light is on otherwise "no" , in a single line.

Sample Input

```
3
6241
8191
0
```

Sample Output

```
no
yes
no
```

Sadi Khan

Suman Mahbub

01-04-2001

14. 10161 - Ant on a Chessboard

Time limit: 3.000 seconds

One day, an ant called Alice came to an M*M chessboard. She wanted to go around all the grids. So she began to walk along the chessboard according to this way: (you can assume that her speed is one grid per second)

At the first second, Alice was standing at (1,1). Firstly she went up for a grid, then a grid to the right, a grid downward. After that, she went a grid to the right, then two grids upward, and then two grids to the left…in a word, the path was like a snake.

For example, her first 25 seconds went like this: (the numbers in the grids stands for the time when she went into the grids)

25	24	23	22	21	5
10	11	12	13	20	4
9	8	7	14	19	3
2	3	6	15	18	2
1	4	5	16	17	1

1 2 3 4 5

At the 8^{th} second , she was at (2,3), and at 20^{th} second, she was at (5,4).

Your task is to decide where she was at a given time.(you can assume that M is large enough)

Input

Input file will contain several lines, and each line contains a number N(1<=N<=2*10^9), which stands for the time. The file will be ended with a line that contains a number 0.

Output

For each input situation you should print a line with two numbers (x, y), the column and the row number, there must be only a space between them.

Sample Input

```
8

20

25

0
```

Sample Output

```
2 3

5 4

1 5
```

15. 10195 - The Knights Of The Round Table

Time limit: 3.000 seconds

Problem B: The Knights Of The Round Table

The Problem

King Arthur is planning to build the round table in a new room, but this time he wants a room that have sunlight entering it, so he planned to build a glass roof. He also wishes his round table to shine during the day, specially at noon, so he wants it to be covered totally by the sunlight. But Lancelot wants the glass part of the room roof to be triangular (and nobody knows the reason why, maybe he made a vow or something like that). So, there will be a triangular area in the room which will be all covered by the sunlight at noon and the round table must be build in this area.

Now, King Arthur wants to build the biggest table that he cans such that it fits in the triangular sunlighted area. As he is not very good in geometry, he asked Galahad to help him (Lancelot is very good in geometry, but King Arthur didn't asked Lancelot to help him because he feared that he would come up with another strange suggestion).

Can you help Galahad (since he's not too good with computers) and write a program which gives the radius of the biggest round table that fits in the sunlighted area? You can assume that the round table is a perfect circle.

The Input

There'll be an arbitrary number of rooms. Each room is represented by three real numbers (a, b and c), which stand for the sizes of the triangular sunlighted area. No triangle size will be greater than 1000000 and you may assume that max(a,b,c) <= (a + b + c) / 2. You must read until you reach the end of the file.

The Output

For each room configuration read, you must print the following line:

```
The radius of the round table is: r
```

Where r is the radius of the biggest round table that fits in the sunlighted area, rounded to 3 decimal digits.

Sample Input

```
12.0 12.0 8.0
```

Sample Output

```
The radius of the round table is: 2.828
```

---

16. 10200 - Prime Time

Time limit: 3.000 seconds

Problem G: Prime Time

The Problem

Euler is a well-known matematician, and, among many other things, he discovered that the formula n^2 + n + 41 produces a prime for 0 <= n < 40. For n = 40, the formula produces 1681, which is 41 * 41. Even though this formula doesn't always produce a prime, it still produces a lot of primes. It's known that for n <= 10000000, there are 47,5% of primes produced by the formula!

So, you'll write a program that will output how many primes does the formula output for a certain interval.

The Input

Each line of input will be given two positive integer a and b such that 0 <= a <= b <= 10000. You must read until the end of the file.

The Output

For each pair a,b read, you must output the percentage of prime numbers produced by the formula in this interval (a <= n <= b) rounded to two decimal digits (with round half up rule).

Sample Input

```
0 39
0 40
39 40
1423 2222
```

Sample Output

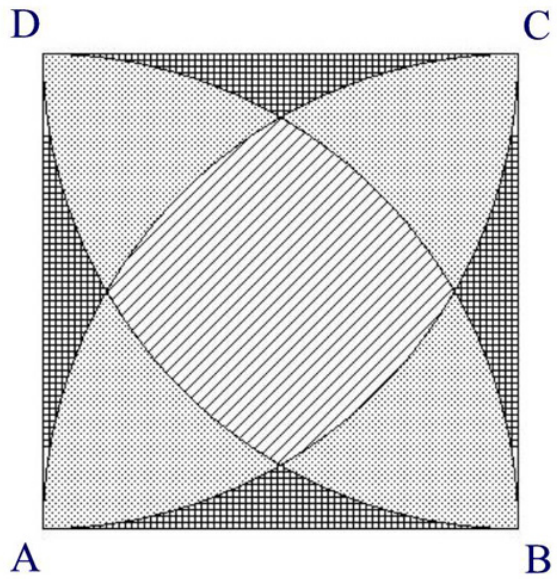
```
100.00
97.56
50.00
44.13
```

---

17. 10209 - Is This Integration ?

Time limit: 3.000 seconds

In the image below you can see a square ABCD, where **AB = BC = CD = DA = a**. Four arcs are drawn taking the four vertexes **A, B, C, D** as centers and **a** as the radius. The arc that is drawn taking **A** as center, starts at neighboring vertex **B** and ends at neighboring vertex **D**. All other arcs are drawn in a similar fashion. Regions of three different shapes are created in this fashion. You will have to determine the total area if these different shaped regions.



Input

The input file contains a floating-point number **a (a>=0 a<=10000)** in each line which indicates the length of one side of the square. Input is terminated by end of file.

Output

For each line of input, output in a single line the total area of the three types of region (filled with different patterns in the image above). These three numbers will of course be floating point numbers with three digits after the decimal point. First number will denote the area of the striped region, the second number will denote the total area of the dotted regions and the third number will denote the area of the rest of the regions.

Sample Input:

```
0.1
0.2
0.3
```

Sample Output:

```
0.003 0.005 0.002
0.013 0.020 0.007
0.028 0.046 0.016
```

Shahriar Manzoor

18. 10221 - Satellites

Time limit: 3.000 seconds

Satellites

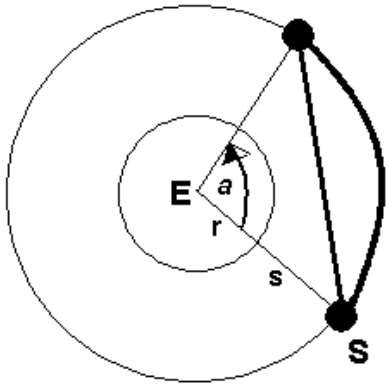
The Problem

The radius of earth is 6440 Kilometer. There are many Satellites and Asteroids moving around the earth. If two Satellites create an angle with the center of earth, can you find out the ***distance*** between them? By ***distance*** we mean both the **arc** and **chord**distances. Both satellites are on the same orbit. (However, please consider that they are revolving on a circular path rather than an elliptical path.)

The Input

The input file will contain one or more test cases.

Each test case consists of one line containing two-integer ***s*** and ***a*** and a string "**min**" or "**deg**". Here ***s*** is the distance of the satellite from the surface of the earth and ***a*** is the angle that the satellites make with the center of earth. It may be in minutes (') or in degrees (^{0}). Remember that the same line will never contain minute and degree at a time.



E = Earth S = Satellite

The Output

For each test case, print one line containing the required distances i.e. both ***arc distance*** and ***chord distance*** respectively between two satellites in Kilometer. The distance will be a floating-point value with six digits after decimal point.

Sample Input

```
500 30 deg
700 60 min
200 45 deg
```

Sample Output

```
3633.775503 3592.408346
124.616509 124.614927
5215.043805 5082.035982
```

Ahmed Shamsul Arefin

19. 10222 - Decode the Mad man

Time limit: 3.000 seconds

Decode the Mad man

The Problem

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

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

The Input

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

The test case consists of one or more words.

The Output

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

Sample Input

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

Sample Output

```
how are you
```

Ahmed Shamsul Arefin

20. 10235 - Simply Emirp

Time limit: 3.000 seconds

Problem G: Simply Emirp

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

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

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

Input

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

Output

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

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

Sample Input

```
17
18
19
179
199
```

Sample Output

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

21. 10242 - Fourth Point !!

Time limit: 3.000 seconds

Problem G
Fourth Point!!
Input: standard input
Output: standard output
Time Limit: 2 seconds
Memory Limit: 32 MB

Given are the **(x, y)** coordinates of the endpoints of two adjacent sides of a parallelogram. Find the **(x, y)** coordinates of the fourth point.

Input

Each line of input contains eight floating point numbers: the **(x, y)** coordinates of one of the endpoints of the first side followed by the **(x, y)** coordinates of the other endpoint of the first side, followed by the **(x, y)** coordinates of one of the endpoints of the second side followed by the **(x, y)** coordinates of the other endpoint of the second side. All coordinates are in meters, to the nearest mm. All coordinates are between**-10000** and **+10000**. Input is terminated by end of file.

Output

For each line of input, print the **(x, y)** coordinates of the fourth point of the parallelogram in meters, to the nearest mm, separated by a single space.

Sample Input

```
0.000 0.000 0.000 1.000 0.000 1.000 1.000 1.000
1.000 0.000 3.500 3.500 3.500 3.500 0.000 1.000
1.866 0.000 3.127 3.543 3.127 3.543 1.412 3.145
```

Sample Output

```
1.000 0.000
-2.500 -2.500
0.151 -0.398
```

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

"You don't get a rank in a contest if you don't solve any problem. So there must be a Problem which can be solved by almost every team."

22. 10252 - Common Permutation

Time limit: 3.000 seconds

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

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

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

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

Sample Input:
```
pretty
women
walking
down
the
street
```

Sample Output:
```
e
nw
et
```

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

23. 10260 - Soundex

Time limit: 3.000 seconds

Problem D: Soundex

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

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

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

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

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

Sample Input

```
KHAWN
PFISTER
BOBBY
```

Output for Sample Input

```
25
1236
11
```

24. 10281 - Average Speed

Time limit: 3.000 seconds

Problem A: Average Speed

You have bought a car in order to drive from Waterloo to a big city. The odometer on their car is broken, so you cannot measure distance. But the speedometer and cruise control both work, so the car can maintain a constant speed which can be adjusted from time to time in response to speed limits, traffic jams, and border queues. You have a stopwatch and note the elapsed time every time the speed changes. From time to time you wonder, "how far have I come?". To solve this problem you must write a program to run on your laptop computer in the passenger seat.

Standard input contains several lines of input: Each speed change is indicated by a line specifying the elapsed time since the beginning of the trip (hh:mm:ss), followed by the new speed in km/h. Each query is indicated by a line containing the elapsed time. At the outset of the trip the car is stationary. Elapsed times are given in non-decreasing order and there is at most one speed change at any given time.

For each query in standard input, you should print a line giving the time and the distance travelled, in the format below.

Sample Input

```
00:00:01 100
00:15:01
00:30:01
01:00:01 50
03:00:01
03:00:05 140
```

Output for Sample Output

```
00:15:01 25.00 km
00:30:01 50.00 km
03:00:01 200.00 km
```

25. 10282 - Babelfish

Time limit: 3.000 seconds

Problem C: Babelfish

You have just moved from Waterloo to a big city. The people here speak an incomprehensible dialect of a foreign language. Fortunately, you have a dictionary to help you understand them.

Input consists of up to 100,000 dictionary entries, followed by a blank line, followed by a message of up to 100,000 words. Each dictionary entry is a line containing an English word, followed by a space and a foreign language word. No foreign word appears more than once in the dictionary. The message is a sequence of words in the foreign language, one word on each line. Each word in the input is a sequence of at most 10 lowercase letters. Output is the message translated to English, one word per line. Foreign words not in the dictionary should be translated as "eh".

Sample Input

```
dog ogday
cat atcay
pig igpay
froot ootfray
loops oopslay

atcay
ittenkay
oopslay
```

Output for Sample Input

```
cat
eh
loops
```

26. 10286 - Trouble with a Pentagon

Time limit: 3.000 seconds

You are asked to place the largest possible square inside a regular pentagon (whose internal angles are same and all the sides are same in length). You are given the information that one vertex of the square will be coincident with a vertex of the square as shown in the figure below. You will have to find the length of a side of the square when a side of the regular pentagon is given.

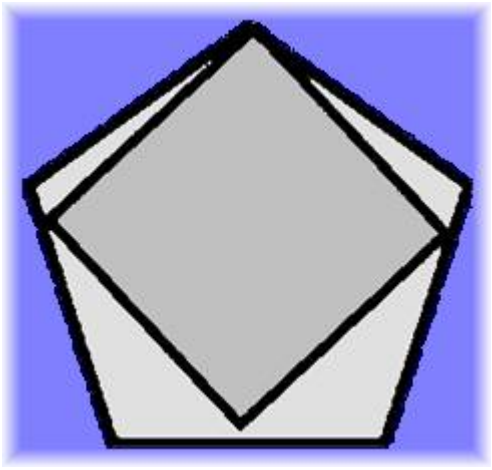


Fig: Square in a pentagon.

Input

The input file contains several lines of input. Each line contains a floating point number**F (0<=F<=100000)** which indicates the length of a side of the pentagon. Input is terminated by end of file.

Output

For each line of input produce one line of output containing a floating point number with ten digits after the decimal point. This number indicates the largest possible side of a square that fits in the pentagon. This output will be judged with a special correction program, so don't worry about small precision errors.

Sample Input

```
0.0000001
0.0000002
0.0000003
```

Sample Output

```
0.0000001067
0.0000002135
0.0000003202
```

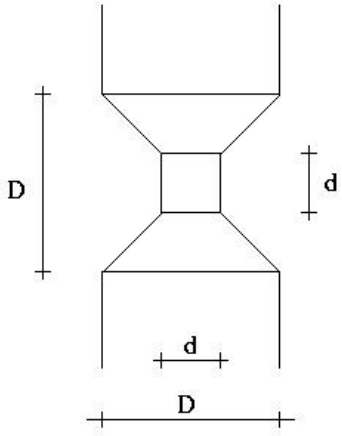
(World Finals Warm-up Contest, Problem Setter: Shahriar Manzoor)

27. 10297 - Beavergnaw

Time limit: 3.000 seconds

Problem C: Beavergnaw

When chomping a tree the beaver cuts a very specific shape out of the tree trunk. What is left in the tree trunk looks like two frustums of a cone joined by a cylinder with the diameter the same as its height. A very curious beaver tries not to demolish a tree but rather sort out what should be the diameter of the cylinder joining the frustums such that he chomped out certain amount of wood. You are to help him to do the calculations.



We will consider an idealized beaver chomping an idealized tree. Let us assume that the tree trunk is a cylinder of diameter **D** and that the beaver chomps on a segment of the trunk also of height**D**. What should be the diameter **d** of the inner cylinder such that the beaver chmped out **V** cubic units of wood?

Input contains multiple cases each presented on a separate line. Each line contains two integer numbers **D** and **V**separated by whitespace. **D** is the linear units and **V** is in cubic units. **V** will not exceed the maximum volume of wood that the beaver can chomp. A line with**D**=0 and **V**=0 follows the last case.

For each case, one line of output should be produced containing one number rounded to three fractional digits giving the value of**d** measured in linear units.

Sample input

```
10 250
20 2500
25 7000
50 50000
0 0
```

Output for sample input

```
8.054
14.775
13.115
30.901
```

28. 10300 - Ecological Premium

Time limit: 3.000 seconds

German farmers are given a premium depending on the conditions at their farmyard. Imagine the following simplified regulation: you know the size of each farmer's farmyard in square meters and the number of animals living at it. We won't make a difference between different animals, although this is far from reality. Moreover you have information about the degree the farmer uses environment-friendly equipment and practices, expressed in a single integer greater than zero. The amount of money a farmer receives can be calculated from these parameters as follows. First you need the space a single animal occupies at an average. This value (in square meters) is then multiplied by the parameter that stands for the farmer's environment-friendliness, resulting in the premium a farmer is paid per animal he owns. To compute the final premium of a farmer just multiply this premium per animal with the number of animals the farmer owns.

Input

The first line of input contains a single positive integer **n (<20)**, the number of test cases. Each test case starts with a line containing a single integer **f (0<f<20)**, the number of farmers in the test case. This line is followed by one line per farmer containing three positive integers each: the size of the farmyard in square meters, the number of animals he owns and the integer value that expresses the farmer's environment-friendliness. Input is terminated by end of file. No integer in the input is greater than **100000** or less than **0**.

Output

For each test case output one line containing a single integer that holds the summed burden for Germany's budget, which will always be a whole number. Do not output any blank lines.

Sample Input

```
3
5
1 1 1
2 2 2
3 3 3
2 3 4
8 9 2
3
9 1 8
6 12 1
8 1 1
3
10 30 40
9 8 5
100 1000 70
```

Sample Output

```
38
86
7445
```

(The Joint Effort Contest, Problem setter: Frank Hutter)

29. 10302 - Summation of Polynomials

Time limit: 3.000 seconds

The following text was taken from a book of mathematics:

The antidifference of a function **f(x)** is the function **g(x)** such that **f(x) = g(x+1)-g(x)**. So, if we have a summation of **f(x)**, it can be simplified by the use of its antidifference in the following way:

f(k)+f(k+1)+f(k+2)+...+f(k+n) =
g(k+1)-g(k)+g(k+2)-g(k+1)+g(k+3)-g(k+2)+...+g(k+n+1)-g(k+n) =
g(k+n+1)-g(k)

A factorial polynomial is expressed as **k^{n}** meaning the following expression: **k * (k-1) * (k-2) * (k-(n-1))**. The antidifference of a factorial polynomial **k^{n}** is**k^{n+1}/(n+1)**.

So, if you want to calculate **S_n = p(1)+p(2)+p(3)+...+p(n)**, where p(i) is a polynomial of degree **k**, we can express **p(i)** as a sum of various factorial polynomials and then, find out the antidifference **P(i)**. So, we have **S_n = P(n+1) - P(1)**.

Example:

S = 2*3 + 3*5 + 4*7 + 5*9 + 6*11 + ... + (n+1)*(2n+1) =
p(1) + p(2) + p(3) + p(4) + p(5) + ... + p(n), where p(i) = (i+1)(2i+1)

Expressing **p(i)** as a factorial polynomial, we have:
p(i) = 2(i)^{2} + 5i + 1.

P(i) = (2/3) (i)^{3} + (5/2) (i)^{2} + i. Calculating **P(n+1) - P(1)** we have

S = (n/6) * (4n^2 + 15n + 17)

Given a number **1 <= x <= 50,000**, one per line of input, calculate the following summation:

1 + 8 + 27 + ... + x^3

Input and Output

Input file contains several lines of input. Each line contain a single number which denotes the value of **x**. Input is terminated by end of file.

For each line of input produce one line of output which is the desired summation value.

Sample Input

```
1
2
3
```

Sample Output

```
1
9
36
```

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

Time limit: 3.000 seconds

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

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

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

This definition can also be converted to an iterative one.

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

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

Output
For each line of input you should output a single line. This line will contain a single integer **n!** if the value of **n!** fits within the unsigned long integer of Arif's computer. Otherwise the line will contain one of the following two words
Overflow! //(When n! > 6227020800)
Underflow! //(When n! < 10000)

Sample Input
```
2
10
100
```

Sample Output
```
Underflow!
3628800
Overflow!
```

(The Decider Contest, Problem setter: Shahriar Manzoor)

31. 10324 - Zeros and Ones

Time limit: 3.000 seconds

Given a string of **0's** and **1's** up to **1000000** characters long and indices **i** and **j**, you are to answer a question whether all characters between position **min(i,j)** and position**max(i,j)** (inclusive) are the same.

Input

There are multiple cases on input. The first line of each case gives a string of **0's** and**1's**. The next line contains a positive integer **n** giving the number of queries for this case. The next **n** lines contain queries, one per line. Each query is given by two non-negative integers, **i** and **j**. For each query, you are to print **Yes** if all characters in the string between position **min(i,j)** and position **max(i,j)** are the same, and **No** otherwise.

Output

Each case on output should start with a heading as in the sample below. The input ends with an empty string that is a line containing only the new line character, this string should not be processed. The input may also with end of file. So keep check for both.

Sample Input

```
0000011111
3
0 5
4 2
5 9
01010101010101010101010101111111111111111111111111111111111111000000000
0000000
5
4 4
25 60
1 3
62 76
24 62
1
1
0 0
```

Sample Output

```
Case 1:
No
Yes
Yes
Case 2:
Yes
Yes
No
Yes
No
Case 3:
Yes
```

32. 10327 - Flip Sort

Time limit: 3.000 second

Flip Sort

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

The Problem

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

The Input

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

The Output

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

Sample Input

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

Sample Output

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

Md. Kamruzzaman

33. 10340 - All in All

Time limit: 3.000 seconds

Problem E
All in All
Input: standard input
Output: standard output
Time Limit: 2 seconds
Memory Limit: 32 MB

You have devised a new encryption technique which encodes a message by inserting between its characters randomly generated strings in a clever way. Because of pending patent issues we will not discuss in detail how the strings are generated and inserted into the original message. To validate your method, however, it is necessary to write a program that checks if the message is really encoded in the final string.

Given two strings *s* and *t*, you have to decide whether *s* is a subsequence of *t*, i.e. if you can remove characters from *t* such that the concatenation of the remaining characters is *s*.

Input Specification

The input contains several testcases. Each is specified by two strings *s*, *t* of alphanumeric ASCII characters separated by whitespace. Input is terminated by EOF.

Output Specification

For each test case output, if *s* is a subsequence of *t*.

Sample Input

```
sequence subsequence
person compression
VERDI vivaVittorioEmanueleReDiItalia
caseDoesMatter CaseDoesMatter
```

Sample Output

```
Yes
No
Yes
No
```

Source: ULM Local Contest

34. 10346 - Peter's Smokes

Time limit: 3.000 seconds

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

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

How many cigarettes can Peter have?

Input

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

Output

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

Sample Input

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

Sample Output

```
5
14
124
```

Source: University of Alberta Local Contest

35. 10347 - Medians

Time limit: 3.000 seconds

Problem C
Medians
Input: standard input
Output: standard output
Time Limit: 1 second
Memory Limit: 32 MB

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.

Sample Input

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

Sample Output

```
5.196
5.196
```

Shahriar Manzoor

36. 10370 - Above Average

Time limit: 3.000 seconds

Problem D: Above Average

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

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

Sample Input

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

Output for Sample Input

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

37. 10407 - Simple division

Time limit: 3.000 seconds

Problem E: Simple division

Integer division between a dividend **n** and a divisor **d** yields a quotient **q** and a remainder **r**. **q** is the integer which maximizes **q*****d** such that **q*****d** <= **n** and **r**= **n** - **q*****d**.

```
     7262.11...
17 ) 123456.00
   - 119
     44
    - 34
     105
    - 102
      36
     - 34
       20
      - 17
        30
```

For any set of integers there is an integer**d** such that each of the given integers when divided by **d** leaves the same remainder.

Each line of input contains a sequence of nonzero integer numbers separated by a space. The last number on each line is 0 and this number does not belong to the sequence. There will be at least 2 and no more than 1000 numbers in a sequence; not all numbers occuring in a sequence are equal. The last line of input contains a single 0 and this line should not be processed. For each line of input, output the largest integer which when divided into each of the input integers leaves the same remainder.

Sample input

```
701 1059 1417 2312 0
14 23 17 32 122 0
14 -22 17 -31 -124 0
0
```

Output for sample input

```
179
3
3
```

Problem Setter: Piotr Rudnicki

38. 10409 - Die Game

Time limit: 3.000 seconds

Life is not easy. Sometimes it is beyond your control. Now, as contestants of ACM ICPC, you might be just tasting the bitter of life. But don't worry! Do not look only on the dark side of life, but look also on the bright side. Life may be an enjoyable game of chance, like throwing dice. Do or die! Then, at last, you might be able to find the route to victory.

This problem comes from a game using a die. By the way, do you know a die? It has nothing to do with "death." A die is a cubic object with six faces, each of which represents a different number from one to six and is marked with the corresponding number of spots. Since it is usually used in pair, "a die" is a rarely used word. You might have heard a famous phrase "the die is cast," though.

When a game starts, a die stands still on a flat table. During the game, the die is tumbled in all directions by the dealer. You will win the game if you can predict the number seen on the top face at the time when the die stops tumbling.

Now you are requested to write a program that simulates the rolling of a die. For simplicity, we assume that the die neither slips nor jumps but just rolls on the table in four directions, that is, north, east, south, and west. At the beginning of every game, the dealer puts the die at the center of the table and adjusts its direction so that the numbers one, two, and three are seen on the top, north, and west faces, respectively. For the other three faces, we do not explicitly specify anything but tell you the golden rule: the sum of the numbers on any pair of opposite faces is always seven.

Your program should accept a sequence of commands, each of which is either "north", "east", "south", or "west". A "north" command tumbles the die down to north, that is, the top face becomes the new north, the north becomes the new bottom, and so on. More precisely, the die is rotated around its north bottom edge to the north direction and the rotation angle is 90 degrees. Other commands also tumble the die accordingly to their own directions. Your program should calculate the number finally shown on the top after performing the commands in the sequence. Note that the table is sufficiently large and the die never falls off during the game.

Input

The input consists of one or more command sequences, each of which corresponds to a single game. The first line of a command sequence contains a positive integer, representing the number of the following command lines in the sequence. You may assume that this number is less than or equal to 1024. A line containing a zero indicates the end of the input. Each command line includes a command that is one of `north`,`east`, `south`, and `west`. You may assume that no white space occurs in any lines.

Output

For each command sequence, output one line containing solely the number on the top face at the time when the game is finished.

Sample Input

```
1
north
3
north
east
south
0
```

Output for the Sample Input

```
5
1
```

39. 10420 - List of Conquests

Time limit: 3.000 seconds

Problem B

List of Conquests

Input: standard input

Output: standard output

Time Limit: 2 seconds

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

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

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

Input

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

Output

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

Sample Input

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

Sample Output

```
England 1
Spain 2
```

40. 10424 - Love Calculator

Time limit: 3.000 seconds

One day I asked Saima that how much she loves me. Her answer was "71.43 %". I was surprised as well as shocked by her answer. I could not understand why she didn't tell 100% and why she told a particular and peculiar fraction like 71.43. Looking at my surprised, shocked and nervous face she burst out laughing and told that she loves me more than any thing in this universe and it was nothing but a silly and funny love calculation. Then she described me the calculation. In this problem you will have to write a program so that any one can calculate love between any two persons very quickly (of course a very silly game).

Rules

You will be given two names. These two names can have white space or some other non-alphabetical characters like $ @ & % etc. But only the alphabets from a to z or A to Z will participate in love calculation. Each alphabet has a particular value. The values are from 1 to 26 in ascending order of the alphabets. Its like this, a = 1, b = 2, c = 3..... z = 26. Both upper case and lower case holds the same values. Then make the sum of these numbers until it comes in one digit.[For example, consider a name 'bcz'. Here, b = 2, c = 3 & z = 26. So, the sum is (2+3+26) = 31 = (3+1) = 4.] Then the ratio of these two numbers in percentage will be the result.

Remember : Result can not be more than 100 % . Take the ratio carefully to avoid this problem.

Input

Your input will be two names. Each name holds not more than 25 characters. End of file will indicate the end of input.

Output

For each pair of names your program will have to calculate the love between those two persons and give the result as output. In result two digits to be displayed after the decimal point. All the results must be in new lines.

Sample input

```
saima

shanto

Pakistan

India

USA

USSR
```

Sample output

```
71.43 %

100.00 %

100.00 %
```

Problem setter: Niaz Morshed Chowdhury (Shanto), East West University, Dhaka

41. 10432 - Polygon Inside A Circle

Time limit: 3.000 seconds

The Problem

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

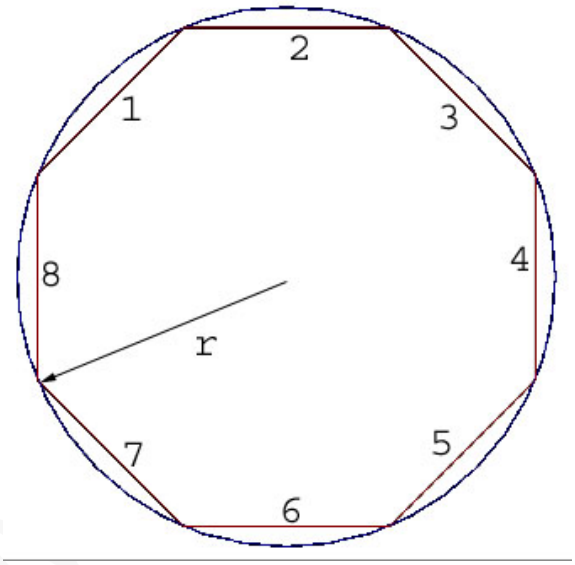


Figure: The regular polygon inside a circle

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

The 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'.

The 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
```

42. 10451 - Ancient Village Sports

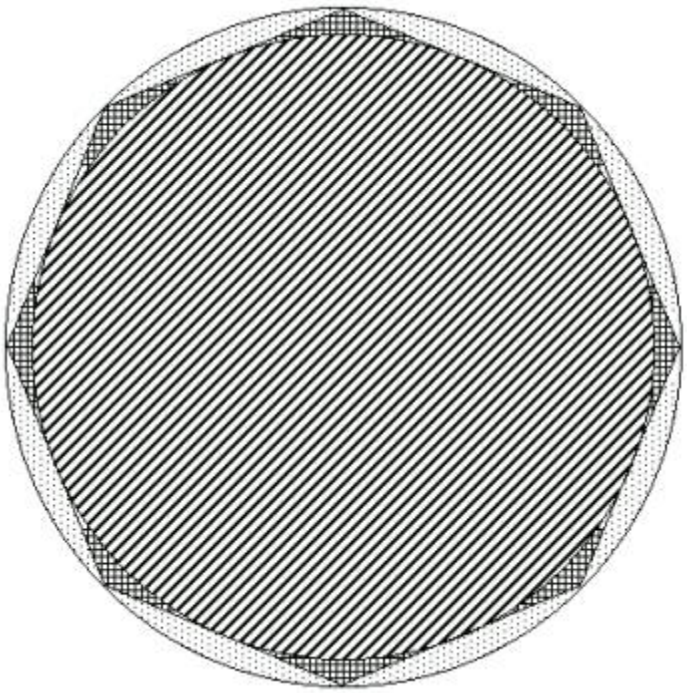
Time limit: 3.000 seconds

Problem H

Ancient Village Sports

Input: standard input
Output: standard output
Time Limit: 30 seconds

In an ancient village there lived a number of people who liked different sorts of sports very much. But there was no built-in sports-ground. So they started looking for places. Ultimately what they found were the surfaces of the rocks with a shape of a regular polygon. Since the spectators are the part and parcel of the game, they must have some place over the ground. Again, officials of the teams must also have some space. They chose the in circle or striped portion of the regular polygon as the playing ground. They decided to leave the dotted portion as indicated in the figure for spectators and the criss-crossed portion for the officials. The diagonally striped portion is used for playing the game. You are going to help them to find the area of these portions.



A polygon is **regular** if all its sides are equal and all its angles are equal. Either of the conditions implies the other in the case of a triangle, but not in general. A rhombus has equal sides but not necessarily equal angles, and a rectangle has equal angles but not necessarily equal sides. So rhombus and rectangle are not regular polygons.

You are given the area (A) of an n-sided regular polygon. You are to determine the total area for the spectators and the total area for the officials. Assume that $\pi = 2*\cos^{-1}(0)$

Input

In each line of the input file there is an integer *n* (0<n<=50) and a floating-point numberA (0<=A<=30000). A line with the value of n is less than three, terminates the input.

Output

For each line of input (except the last one) you should produce one line of output. This line contains the serial no of output as shown in the sample output followed by two floating-point numbers separated by a single space. The first one gives the area for the spectators and the second one gives that of the officials. There is also a single space between the colon and first floating point number. The floating-point numbers has five digits after the decimal point.

Sample Input

```
3 0.43301

6 2.59808

9 6.18182

0 2.33333
```

Sample Output

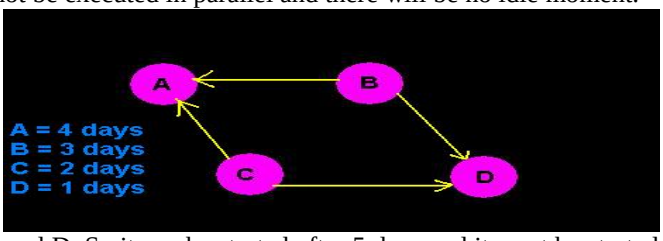
```
Case 1: 0.61418 0.17121

Case 2: 0.54352 0.24188

Case 3: 0.53226 0.25314
```

Problem-setter: Tanbir Ahmed, CSE Department, Southeast University

43. 10461 - Difference

Time limit: 3.000 seconds

You are given a list of jobs with associated time to complete them. Also are given a list of job dependencies. Your job is to calculate the maximum possible difference among all possible completion time of a particular job. For this problem, job dependency is a bit simpler. When a job depends on several other jobs to be completed, finishing of all of them will allow it to execute. Furthermore, jobs cannot be executed in parallel and there will be no idle moment.



In the above example job-B cannot be started before A and D. So it can be started after 5 days and it must be started after 7 days. Here for job-B, the difference is 2 days.

Input

Input starts with a pair of integers **v(1<=v<=500)** and **e(0<=e<=500)** where **v**represents the number of jobs and **e** represents the number of dependencies in the dependency list. Following is a line with **v** integers each indicating the time in days to complete the jobs where the i'th integer denotes the completion time of the i'th job. The jobs are numbered by integers from 1 to **v**. Next **e** lines has the form "**x y**" which means that job **x** should be completed prior to job **y(1 <= x, y <= v)**. Next line has an integer **q** which denotes the number of queries to answer which is followed by **q**integers **x (1 <= x <= v)**. A line with **v** = **e** = 0 ends the input session. Every block will be followed by a blank line. There will be no impractical situation (Each job can be completed) in input.

Output

For each query in a block output the result according to the problem description in a separate line. A blank line is essential after every block of data. See the sample output.

Sample Input

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

4 4
4 3 2 1
2 1
2 4
3 1
3 4
2
3 4

0 0
```

Sample Output

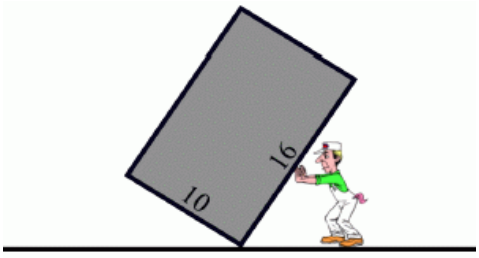
```
Case #1:
1
2

Case #2:
3
4
```

44. 10473 - Simple Base Conversion

Time limit: 3.000 seconds

Simple Base Conversion

In this problem you are asked to write a simple base conversion program. You will be given a hexadecimal or decimal integer number as input. You will have to output the corresponding decimal or hexadecimal number. Hexadecimal numbers always starts with a `0x' and all other numbers are to be considered as decimal numbers. There will be no invalid numbers in the input.



Input

The input file contains several lines of input. Each line contains a single non-negative number, which may be a decimal or hexadecimal number as explained in the problem statement. The decimal value of this number will be less than 2^{31}. A line containing a negative decimal number terminates input. This number should not be processed. Input numbers will contain no space within them.

Output

For each line of input (Except the last one) produce one line of output. This line should contain the decimal or hexadecimal representation of the corresponding hexadecimal or decimal number. Like the input, the hexadecimal numbers in the output should be preceded by a `0x'.

Sample Input

```
4
7
44
0x80685
-1
```

Sample Output

```
0x4
0x7
0x2C
525957
```

45. 10490 - Mr. Azad and his Son!!!!!

Time limit: 3.000 seconds

There are a lot of Abul Kalam Azad in Bangladesh. But, why is he so special? Not that he is my dad is the only reason. He can wonderfully do some calculation. If anyone gives him any positive integer, he amazingly can say the relative perfect number using the formula 2^(k-1)*(2^k-1) without using neither calculator nor computer. Say, I have told him to find out the relative perfect number of 2, he replies 6 which is a perfect number. But perfect is not possible for all the integers. I have asked him the process, but he says that I should find this thing out by myself how an integer is related to a perfect number. Anyway, I have challenged him that it is very possible for me to do the same calculation using a computer. Although I could not figure out how he can do this, I know that the next ACM Online Programming Contest is near at hand and World's top programmers are available to solve my very simple problem.

Now, you are to write a program for me to win over my dad, which will take input **n**, and determine the perfect number **p**.

Input

An integer **1<n<=31** is given in each input line. Input is terminated by a zero in a single line. This input should not be processed. All the output numbers will fit in 64 bit signed integer.

Output

Output will be in the following format:

If perfect number is possible -
Perfect: p!

If perfect number is not possible, but given number is prime -
Given number is prime. But, NO perfect number is available.

If perfect number is not possible and given number is not prime -
Given number is NOT prime! NO perfect number is available.

Sample Input

```
2
3
6
0
```

Sample Output

```
Perfect: 6!
Perfect: 28!
Given number is NOT prime! NO perfect number is available.
```

46. 10499 - The Land of Justice

Time limit: 3.000 seconds

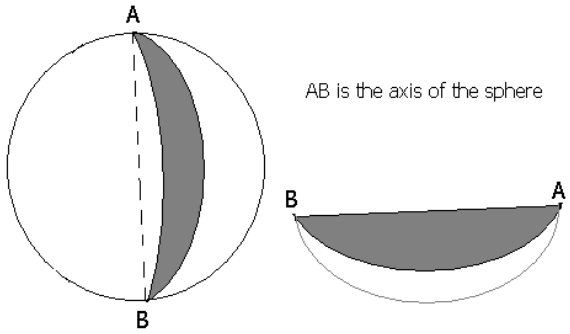
In the Land of Justice the selling price of everything is fixed all over the country. Nobody can buy a thing and sell it in double price. But, that created problems for the businessmen. They left their business and went to the production. So, after some days everybody was in production and nobody in business. And the people didn't get their necessary things though the country was self-sufficient in every sector.

The government became very much anxious. But, they were intelligent enough to call the mathematicians.

The mathematicians gave a solution. They suggested setting the surface area of an object as its selling-unit instead of its volume. Actually the clever mathematicians were very much interested to establish their own business.

Now, the government asks the programmers to build the software that would calculate the profit things.

Here your job is to calculate the business profit for a solid sphere. A businessman buys a complete sphere and to maximize his profit he divides it in **n** equal parts. All cut should go through the axis of the sphere. And every part should look like the picture below:



Input

You are given a sequence of integers N ($0 < N < 2^{31}$), indicating the numbers of parts of the sphere. The input file is terminated with a negative number. This number should not be processed.

Output

Calculate the profit over the sold pieces. The result should be in percentage and rounded to the nearest integer.

Sample input

```
2

2

-1
```

Sample output

```
50%

50%
```

47. 10515 - Powers Et Al.

Time limit: 3.000 seconds

Problem G
Power et al.
Input: Standard Input

Output: Standard Output

Finding the exponent of any number can be very troublesome as it grows exponentially☺. But in this problem you will have to do a very simple task. Given two non-negative numbers **m** and **n**, you will have to find the last digit of $\mathbf{m^n}$ in decimal number system.

Input

The input file contains less than **100000** lines. Each line contains two integers **m** and **n**(Less than **10^101**). Input is terminated by a line containing two zeroes. This line should not be processed.

Output

For each set of input you must produce one line of output which contains a single digit. This digit is the last digit of $\mathbf{m^n}$.

Sample Input	Output for Sample Input
2 2 2 5 0 0	4 2

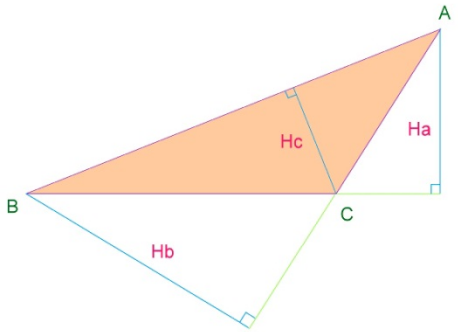
Problemsetter: Shahriar Manzoor

48. 10522 - Height to Area

Time limit: 3.000 seconds

Problem

It's an easy geometry problem. For any triangle ABC we know that the height from A to the line BC (or it's extension) is Ha, from B to the line AC (or it's extension) is Hb and from C to the line AB (or it's extension) is Hc. Now you are given these three values and you have to figure out the area of the ABC .



Input

At first the input will be an integer n. Which denotes the number of invalid inputs after which the input will terminate. Then there will be three real numbers Ha, Hb and Hc per line.

Output

For each input block there should be one output line. For valid inputs the line contains the area of the ABC up to 3 decimal places after the decimal point and for invalid inputs there will be a line "These are invalid inputs!". After n invalid input sets the program will terminate.

Sample Input

```
1
31.573 22.352 63.448
46.300 50.868 86.683
22.005 24.725 22.914
5.710 25.635 32.805
```

Sample Output

```
1517.456
2219.941
311.804
These are invalid inputs!
```

49. 10533 - Digit Primes

Time limit: 3.000 seconds

A prime number is a positive number, which is divisible by exactly two different integers. A digit prime is a prime number whose sum of digits is also prime. For example the prime number **41** is a digit prime because **4+1=5** and **5** is a prime number.**17** is not a digit prime because **1+7 = 8**, and **8** is not a prime number. In this problem your job is to find out the number of digit primes within a certain range less than**1000000**.

Input

First line of the input file contains a single integer **N (0<N<=500000)** that indicates the total number of inputs. Each of the next **N** lines contains two integers **t1** and **t2(0<t1<=t2<1000000).**

Output

For each line of input except the first line produce one line of output containing a single integer that indicates the number of digit primes between **t1** and **t2** (inclusive).

Sample Input

3 10 20 10 100 100 10000	1 10 576

Problemsetter: Shahriar Manzoor, Member of Elite Problemsetters' Panel

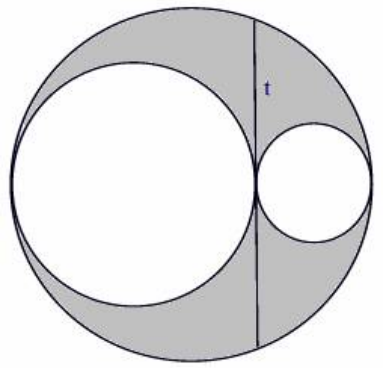
/*Note: You should at least use scanf() and printf() to take input and produce output for this problem. cin and cout is too slow for this problem to get it within time limit.*/

50. 10573 - Geometry Paradox

Time limit: 3.000 seconds

Time Limit: 1 Second

In the picture below you can see two small circles touching each other. The larger circle touches both pf them. The length of the common tangent inscribed inside the larger circle is t and the radiuses of the two smaller circles are **r1** and **r2**. The centers of the three circles are collinear. You will be given the value of **r1** and **r2** or the value of **t**. You will have to find the area that is within the larger circle but out of the two smaller circles (marked gray in the picture). If the given data is not enough to find the gray area, print the line **"Impossible."** in a single line



Input

First line of the input file contains an integer **N (N<=100)** which denotes how many sets of inputs are there. Each of the next **N** lines contain a set of input.

Each set either contains one or two integer. If it contains one integer then it is the value of **t**, otherwise the two values are the values of **r1** and **r2**. All these integers are less than **100**

Output

For each line of input produce one line of output. This line contains the area of the gray part if the given information is enough to find the area of the gray part. Otherwise it contains the line "Impossible." The area should have four digits after the decimal point. Assume that pi=**2*cos^{-1}(0)**.

Sample Input	Output for Sample Input
2 10 10 15 20	628.3185 1884.9556

Problemsetter: Shahriar Manzoor, Member of Elite Problemsetters' Panel

51. 10591 - Happy Number

Time limit: 3.000 seconds

Problem C	Happy Number
Time Limit	1 Second

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

Input

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

Output

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

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

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

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

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

52. 10611 - The Playboy Chimp

Time limit: 1.000 seconds

Once upon a time, there lived a chimpanzee called *Luchu Bandor (aka Playboy Chimp)*. Luchu was unhappily married to *Bunty Mona*, a short but cute little lady chimp. Luchu was tall and handsome – he was feeling uncomfortable taking Bunty to public places along with him. People would stare at them all the while. At one point, Luchu could not stand it anymore and he decided to do some justice to his name. He started looking for *a new hope* in the Lady Chimps' High School. Every day Luchu would climb up a bamboo tree and wait for the morning drill to start. From there he could see each and every lady chimp doing their routine drill. Now, Luchu was looking for the tallest lady chimp that would be shorter than he; he would also like to consider someone a little taller than he. But someone of his same height will never be on his list. Every morning Luchu picks up a line of lady chimps and finds the best two according to his set criterion. His job has been made easy by the fact that the lady chimps in each line are ordered by their height, the shortest one is in the front and the tallest one is at the back. Your task is to help Luchu on one particular day to find two lady chimps: the tallest one shorter than he and the shortest one taller than he.

Input

There will be only one set of input for this problem. The first line of input gives you a number **N (1≤N≤50000)**, the number of lady chimps on the line. In the next line you would have **N** integers (in the range **1** to 2^{31}**-1**) giving the heights of the **N** chimps. There would be a single space after every number. You can assume that the chimps are ordered in non-decreasing order of their heights. In the next line you would have an integer **Q (1≤Q≤25000)** giving the number of queries. Then in the next line **Q** queries will follow. Then you would have **Q** numbers giving the height of Luchu! Don't worry, Luchu is from the land where people can have **3** birthdates; **Q** heights for a chimpanzee will make no difference here. The **Q** numbers are listed on a line and their range from **1 to** 2^{31}**-1**, and as before you would find a single space after every query number. The query numbers are not supposed to come in any particular order.

Output

For each query height, print two numbers in one line. The first one would be the height of the tallest lady chimp that is shorter than Luchu, and the next number would be the height of the shortest lady chimp that is taller than he. These two numbers are to be separated by a single space. Whenever it is impossible to find any of these two heights, replace that height with an uppercase **'X'**.

Sample Input	Output for Sample Input
4 1 4 5 7 4 4 6 8 10	1 5 5 7 7 X 7 X

53. 10678 - The Grazing Cow

Time limit: 3.000 seconds

Time Limit: 1 second

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<=100)**, which indicates how many sets of inputs are there. Each of the next **N** lines contains two integers **D(0<=D<=1000)**and **L(D<L<=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	Output for Sample Input
3 10 12 23 45 12 18	62.517 1366.999 189.670

54. 10679 - I Love Strings!!

Time limit: 3.000 seconds

I love Strings!!!

Input / Output: standard I/O

Time Limit: 4 sec

Hmmmmmm……strings again :) Then it must be an easy task for you. Well……you are given with a string **S** of length not more than 100,000 characters (only 'a'-'z' and 'A' – 'Z'). Then follows **q** (*q < 1000*) queries where each query contains a string **T** of maximum length 1,000 (also contains only 'a'-'z' and 'A' – 'Z'). You should determine whether or not **T** is a substring of **S**.

Input

First line contains an integer **k** (*k < 10*) telling the number of test cases to follow. Each test case begins with **S**. It is followed by **q**. After this line there are **q** lines each of which has a string **T** as defined before.

Output

For each query print 'y' if it is a substring of **S** or 'n' otherwise followed by a new line. See the sample output below.

Sample Input

```
2
abcdefghABCDEFGH
2
abc
abAB
xyz
1
xyz
```

Sample Output

```
y
n
y
```


---

55. 10683 - The decadary watch

Time limit: 3.000 seconds

Background

In the first years following the French Revolution, intellectuals were set to outroot the society from the traditions and superstitions the dark ages of the royalty. Some of these contributions have had a worldwide success, such as the metric system. Others have falled into (almost) complete oblivion, such as the decimal clock system, invented by the mathematician Gilbert Romme. The decimal clock system divides the day in 10 decimal hours, themselves divided in 100 decimal minutes, themselves divided into 100 decimal seconds.

The Problem

You are commissioned by the international watch maker "Splatch" to include yet another useless feature in their next line of product: decimal time display. Your first task will be to implement a program that converts a traditional time into a decimal time at the precision of one-hundredth of second.

Input

The input consists of a sequence of lines, each containing exactly one traditional time, in the format `HHMMSSCC`, where **0 ≤ HH ≤ 23**, **0 ≤ MM ≤ 59**, **0 ≤ SS ≤ 59** and **0 ≤ CC ≤ 99**. The input is terminated with an end-of-file.

Output

For each given traditional time, the output will echo a line with the corresponding decimal time, rounded by truncation, in the format `HMMSSCC`, where **0 ≤ H ≤ 9**, **0 ≤ MM ≤ 99**, **0 ≤ SS ≤ 99** and **0 ≤ CC ≤ 99**.

Sample input

```
00000000
23595999
12000000
14273467
02475901
```

Sample output

```
0000000
9999998
5000000
6024846
1166552
```

56. 10689 - Yet another Number Sequence

Time limit: 3.000 seconds

Problem B
Yet another Number Sequence
Input: standard input
Output: standard output
Time Limit: 3 seconds

Let's define another number sequence, given by the following function:

```
f(0) = a
f(1) = b
f(n) = f(n-1) + f(n-2), n > 1
```

When **a = 0** and **b = 1**, this sequence gives the Fibonacci Sequence. Changing the values of **a** and **b** , you can get many different sequences. Given the values of **a, b**, you have to find the last **m** digits of **f(n)** .

Input

The first line gives the number of test cases, which is less than **10001**. Each test case consists of a single line containing the integers **a b n m**. The values of **a** and **b** range in**[0,100]**, value of **n** ranges in **[0, 1000000000]** and value of **m** ranges in **[1, 4]**.

Output

For each test case, print the last **m** digits of **f(n)**. However, you should **NOT** print any leading zero.

Sample Input	Output for Sample Input
4 0 1 11 3 0 1 42 4 0 1 22 4 0 1 21 4	89 4296 7711 946

Problem setter: Sadrul Habib Chowdhury

Special Thanks: Derek Kisman, Member of Elite Problem Setters' Panel

57. 10696 - f91

Time limit: 3.000 seconds

Problem A - f91

Time Limit: 1 second

Background

McCarthy is a famous theorician of computer science. In his work, he defined a recursive function, called f91, that takes as input a positive integer *N* and returns a positive integer defined as follows:

- If $N \leq 100$, then f91(*N*) = f91(f91(*N*+11));
- If $N \geq 101$, then f91(*N*) = *N*-10.

The Problem

Write a program, that computes McCarthy's f91.

The Input

The input tests will consist of a series of positive integers, each integer is at most 1,000,000. There will be at most 250,000 test cases. Each number is on a line on its own. The end of the input is reached when the number 0 is met. The number 0 shall not be considered as part of the test set.

Output

The program shall output each result on a line by its own, following the format given in the sample output.

Sample input

```
500
91
0
```

Sample output

```
f91(500) = 490
f91(91) = 91
```

58. 10699 - Count the factors

Time limit: 3.000 seconds

Problem D - Count the factors

Time Limit: 1 second

The Problem

Write a program, that computes the number of different prime factors in a positive integer.

The Input

The input tests will consist of a series of positive integers. Each number is on a line on its own. The maximum value is 1000000. The end of the input is reached when the number 0 is met. The number 0 shall not be considered as part of the test set.

Output

The program shall output each result on a line by its own, following the format given in the sample output.

Sample input

```
289384
930887
692778
636916
747794
238336
885387
760493
516650
641422
0
```

Sample output

```
289384 : 3
930887 : 2
692778 : 5
636916 : 4
747794 : 3
238336 : 3
885387 : 2
760493 : 2
516650 : 3
641422 : 3
```

59. 10700 - Camel trading

Time limit: 3.000 seconds

Problem E - Camel trading

Time Limit: 1 second

Background

Aroud 800 A.D., El Mamum, Calif of Baghdad was presented the formula 1+2*3*4+5, which had its origin in the financial accounts of a camel transaction. The formula lacked parenthesis and was ambiguous. So, he decided to ask savants to provide him with a method to find which interpretation is the most advantageous for him, depending on whether is is buying or selling the camels.

The Problem

You are commissioned by El Mamum to write a program that determines the maximum and minimum possible interpretation of a parenthesis-less expression.

Input

The input consists of an integer **N**, followed by **N** lines, each containing an expression. Each expression is composed of at most **12** numbers, each ranging between **1** and **20**, and separated by the sum and product operators **+** and *****.

Output

For each given expression, the output will echo a line with the corresponding maximal and minimal interpretations, following the format given in the sample output.

Sample input

```
3
1+2*3*4+5
4*18+14+7*10
3+11+4*1*13*12*8+3*3+8
```

Sample output

```
The maximum and minimum are 81 and 30.
The maximum and minimum are 1560 and 156.
The maximum and minimum are 339768 and 5023.
```

60. 10703 - Free spots

Time limit: 3.000 seconds

Problem H - Free spots

Time Limit: 1 second

The Problem

Write a program, that, given a board, and a list of rectangular sub-portions of the board, returns the number of positions that belong to no sub-portion.

The Input

The input consists of a series of test sets separated by blank lines. A test set starts with a line with three numbers W, H and N, giving respectively the width, the height and the number of sub-boards. These values satisfy the following constraints: $1 \leq W, H \leq 500$ and $0 \leq N \leq 99$. Follow then N lines, composed of four integers X1, Y1, X2, Y2, such that (X1, Y1) and (X2, Y2) are the positions of two opposite corners of a sub-board. These values satisfy the following constraints: $1 \leq X1, X2 \leq W$ and $1 \leq Y1, Y2 \leq H$. The end of the input is reached when the numbers W, H and N are equal to 0. This last line shall not be considered as a test set.

Output

The program shall output each result on a line by its own, following the format given in the sample output.

Sample input

```
1 1 1
1 1 1 1

2 2 2
1 1 1 2
1 1 2 1

493 182 3
349 148 363 146
241 123 443 147
303 124 293 17

0 0 0
```

Sample output

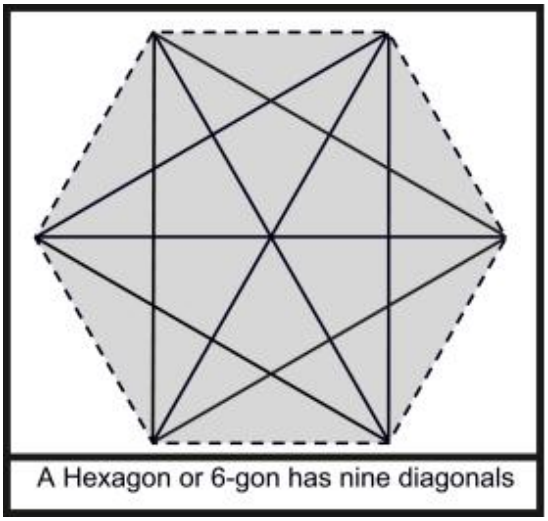
```
There is no empty spots.
There is one empty spot.
There are 83470 empty spots.
```

61. 10784 - Diagonal

Time limit: 3.000 seconds

Diagonal

The number of diagonals of an ***n***-gon is not less than ***N***. What is the minimum possible value of ***n***?



Input

The input file contains less than 1001 lines of inputs. Each line contains a positive integer ***N*** (***N*** $\leq 10^{15}$) that indicates the minimum possible number of diagonals. Input is terminated by a line containing a zero. This line should not be processed.

Output

For each line of input produce one line of output, which contains the output serial number, and also the minimum possible value for ***n*** (Number of sides).

Sample Input

```
10
100
1000
0
```

Sample Output

```
Case 1: 7
Case 2: 16
Case 3: 47
```

Miguel Revilla 2004-12-02

62. 10789 - Prime Frequency

Time limit: 3.000 seconds

Prime Frequency

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

Input

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

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

Output

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

Sample Input

```
3
ABCC
AABBBBDDDDD
ABCDFFFF
```

Sample Output

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

63. 10790 - How Many Points of Intersection?

Time limit: 3.000 seconds

How Many Points of Intersection?

We have two rows. There are ***a*** dots on the top row and ***b*** dots on the bottom row. We draw line segments connecting every dot on the top row with every dot on the bottom row. The dots are arranged in such a way that the number of internal intersections among the line segments is maximized. To achieve this goal we must not allow more than two line segments to intersect in a point. The intersection points on the top row and the bottom are not included in our count; we can allow more than two line segments to intersect on those two rows. Given the value of ***a*** and ***b***, your task is to compute ***P***(***a***, ***b***), the number of intersections in between the two rows. For example, in the following figure ***a*** = 2 and ***b*** = 3. This figure illustrates that ***P***(2, 3) = 3.

Input

Each line in the input will contain two positive integers ***a*** (0 < ***a*** $\leq$ 20000) and ***b*** (0 <***b*** $\leq$ 20000). Input is terminated by a line where both ***a*** and ***b*** are zero. This case should not be processed. You will need to process at most 1200 sets of inputs.

Output

For each line of input, print in a line the serial of output followed by the value of ***P***(***a***,***b***). Look at the output for sample input for details. You can assume that the output for the test cases will fit in **64-bit** signed integers.

Sample Input

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

Sample Output

```
Case 1: 1
Case 2: 3
Case 3: 9
```

64. 10812 - Beat the Spread!

Time limit: 3.000 seconds

Problem D: Beat the Spread!

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

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

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

Sample Input

```
2
40 20
20 40
```

Output for Sample Input

```
30 10
impossible
```

Gordon V. Cormack

65. 10815 - Andy's First Dictionary

Time limit: 3.000 seconds

Andy, 8, has a dream - he wants to produce his very own dictionary. This is not an easy task for him, as the number of words that he knows is, well, not quite enough. Instead of thinking up all the words himself, he has a briliant idea. From his bookshelf he would pick one of his favourite story books, from which he would copy out all the distinct words. By arranging the words in alphabetical order, he is done! Of course, it is a really time-consuming job, and this is where a computer program is helpful.

You are asked to write a program that lists all the different words in the input text. In this problem, a word is defined as a consecutive sequence of alphabets, in upper and/or lower case. Words with only one letter are also to be considered. Furthermore, your program must be CaSe InSeNsItIvE. For example, words like "Apple", "apple" or "APPLE" must be considered the same.

Input

The input file is a text with no more than 5000 lines. An input line has at most 200 characters. Input is terminated by EOF.

Output

Your output should give a list of different words that appears in the input text, one in a line. The words should all be in lower case, sorted in alphabetical order. You can be sure that he number of distinct words in the text does not exceed 5000.

Sample Input

```
Adventures in Disneyland

Two blondes were going to Disneyland when they came to a fork in the
road. The sign read: "Disneyland Left."

So they went home.
```

Sample Output

```
a
adventures
blondes
came
disneyland
fork
going
home
in
left
read
road
sign
so
the
they
to
two
went
were
when
```

66. 10921 - Find the Telephone

Time limit: 3.000 seconds

Time Limit: 1 second

In some places is common to remember a phone number associating its digits to letters. In this way the expression **MY LOVE** means **69 5683**. Of course there are some problems, because some phone numbers can not form a word or a phrase and the digits **1** and **0** are not associated to any letter.

Your task is to read an expression and find the corresponding phone number based on the table below. An expression is composed by the capital letters (**A-Z**), hyphens (-) and the numbers **1** and **0**.

Letters	Number
ABC	2
DEF	3
GHI	4
JKL	5
MNO	6
PQRS	7
TUV	8
WXYZ	9

Input

The input consists of a set of expressions. Each expression is in a line by itself and has**C** characters, where $\mathbf{1 \leq C \leq 30}$. The input is terminated by enf of file (EOF).

Output

For each expression you should print the corresponding phone number.

Sample Input

```
1-HOME-SWEET-HOME
MY-MISERABLE-JOB
```

Sample Output

```
1-4663-79338-4663
69-647372253-562
```

67. 10922 - 2 the 9s

Time limit: 3.000 seconds

Problem C - 2 the 9s
Time Limit: 1 second

Introduction to the problem

A well-known trick to know if an integer N is a multiple of nine is to compute the sum S of its digits. If S is a multiple of nine, then so is N. This is a recursive test, and the depth of the recursion needed to obtain the answer on N is called the 9-degree of N.

Your job is, given a positive number N, determine if it is a multiple of nine and,if it is, its 9-degree.

Description of the input

The input is a file such that each line contains a positive number. A line containing the number 0 is the end of the input. The given numbers can contain up to 1000 digits.

Description of the output

The output of the program shall indicate, for each input number, if it is a multiple of nine, and in case it is, the value of its nine-degree. See the sample output for an example of the expected formatting of the output.

Sample input:

```
99999999999999999999
9
99999999999999999999999999999998
0
```

Sample output

```
99999999999999999999 is a multiple of 9 and has 9-degree 3.
9 is a multiple of 9 and has 9-degree 1.
99999999999999999999999999999998 is not a multiple of 9.
```

---

68. 10924 - Prime Words

Time limit: 3.000 seconds

Problem E - Prime Words

Time Limit: 1 second

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

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

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

Input

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

Output

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

Sample Input

```
UFRN
contest
AcM
```

Sample Output

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

Problem setter: S�rgio Queiroz de Medeiros

69. 10929 - You can say 11

Time limit: 3.000 seconds

Problem C - You can say 11
Time Limite: 1 second

Introduction to the problem

Your job is, given a positive number N, determine if it is a multiple of eleven.

Description of the input

The input is a file such that each line contains a positive number. A line containing the number 0 is the end of the input. The given numbers can contain up to 1000 digits.

Description of the output

The output of the program shall indicate, for each input number, if it is a multiple of eleven or not.

Sample input:

112233
30800
2937
323455693
5038297
112234
0

Sample output

112233 is a multiple of 11.
30800 is a multiple of 11.
2937 is a multiple of 11.
323455693 is a multiple of 11.
5038297 is a multiple of 11.
112234 is not a multiple of 11.

---

70. 10931 - Parity

Time limit: 3.000 seconds

Problem E - Parity

Time Limit: 1 second

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

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

Input

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

Output

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

Sample Input

```
1
2
10
21
0
```

Sample Output

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

Problem setter: S�rgio Queiroz de Medeiros

71. 10970 - Big Chocolate

Time limit: 3.000 seconds

Problem G

Big Chocolate

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 friendsexactly 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.

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.

Figure 1. Mohammad's chocolate

The 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.

The 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

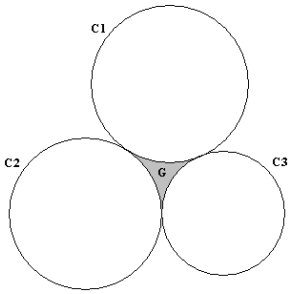
```
2 2
1 1
1 5
```

Sample Output

```
3
0
4
```

72. 10991 - Region

Time limit: 3.000 seconds



From above figure, it is clear that **C1**, **C2** and **C3** circles are touching each other.

Consider,

C1 circle have **R1** radius.

C2 circle have **R2** radius.

C3 circle have **R3** radius.

Write a program that will calculate the area of shaded region **G**

Input

The first line will contain an integer **k (1 ≤ k ≤ 1000)** which is the number of cases to solve. Each of the following k Lines will contain three floating point number **R1 (1 ≤ R1 ≤ 1000)**, **R2 (1 ≤ R2 ≤ 1000)** and **R3 (1 ≤ R3 ≤ 1000)**.

Output

For each line of input, generate one line of output containing the area of **G** rounded to six decimal digits after the decimal point. Floating-point errors will be ignored by special judge program.

Sample Input	Output for Sample Input
2 5.70 1.00 7.89 478.61 759.84 28.36	1.224323 2361.005761