Problem A

寻找素数

Time Limit: 1.000 second Memory Limit: 131072 KB

A prime number is a natural number which has exactly two distinct natural number divisors: 1 and itself.

For example, the first four prime numbers are: 2, 3, 5 and 7.

Write a program which reads a list of N integers and prints the number of prime numbers in the list.

Input

The first line contains an integer N, the number of elements in the list.

N numbers are given in the following lines.

Output

Print the number of prime numbers in the given list.

Constraints

$$1 \leq N \leq 10000$$

$$2 \leq an\ element\ of\ the\ list \leq 10^8$$

Sample Input 1	Sample Output 1
5 2 3 4 5 6	3

Problem B

究竟能得几分?

Time Limit: 3.000 seconds Memory Limit: 65536 KB

There is an objective test result such as “OOXXOXXOOO”. An ‘O’ means a correct answer of a problem and an ‘X’ means a wrong answer. The score of each problem of this test is calculated by itself and its just previous consecutive ‘O’s only when the answer is correct.

For example, the score of the 10th problem is 3 that is obtained by itself and its two previous consecutive ‘O’s.

Therefore, the score of “OOXXOXXOOO” is 10 which is calculated by “1+2+0+0+1+0+0+1+2+3”.

You are to write a program calculating the scores of test results.

Input

Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case starts with a line containing a string composed by ‘O’ and ‘X’ and the length of the string is more than 0 and less than 80. There is no spaces between ‘O’ and ‘X’.

Output

Your program is to write to standard output. Print exactly one line for each test case. The line is to contain the score of the test case.

Sample Input 1	Sample Output 1
5 OOXXOXXOOO OOXXOOXXOO OXOXOXOXOXOXOX OOOOOOOOOO OOOOXOOOOXOOOOX	10 9 7 55 30

Problem C

Super Stairs

Time Limit: 1.000 second   Memory Limit: 23768 KB

有一个楼梯共M级，刚开始时你在第一级，若每次只能跨上一级或二级，要走上第M级，共有多少种走法？

Input

输入数据首先包含一个整数N，表示测试实例的个数，然后是N行数据，每行包含一个整数M（1<=M<=40）表示楼梯的级数。

Output

对于每个测试实例，请输出不同走法的数量

Sample Input 1	Sample Output 1
2	1
2	2
3	

Problem D

谁，亮着?

Time Limit: 1.000 second Memory Limit: 131072 KB

一个房间里有N盏灯，刚开始所有的灯都是关的。编号为 1 的人走过来，把是 1 的倍数的灯全部打开，编号为 2 的人把是 2 的倍数的灯全部关上，编号为 3 的人又把是 3 的倍数的灯开的关上，关的开起来（第 i 个人把编号为 i 的倍数的灯切换状态）……直到第 N 个人操作完为止。

给定 N ， 求 N 轮之后，还有哪几盏是开着的。

Input

一个数字 $N(1 \le N \le 10^{12})$，表示灯的个数和操作的轮数。

Output

若干数，表示开着的电灯编号。

Sample Input	Sample Output
6	1 4

* 输出结果只有一行，且以单个空格作为数字间的分隔

Problem E

Decimal to Binary

Time Limit: 1.000 second Memory Limit: 32768 KB

Give you a number on base ten, you should output it on base two. $(0 < N < 1000)$

Input

For each case there is a positive number N on base ten.

Input is terminated by EOF.

Output

For each case output a number on base two.

Sample Input	Sample Output
1	1
2	10
3	11

Problem F

天生一对难自弃

Time Limit: 3.000 second Memory Limit: 65536 KB

You are given a string consisting of parentheses `()` and `[]`. A string of this type is said to be correct:

(a) if it is the empty string

(b) if A and B are correct, AB is correct,

(c) if A is correct, (A) and [A] is correct.

Write a program that takes a sequence of strings of this type and check their correctness. Your program can assume that the maximum string length is 128.

Input

The file contains a positive integer n and a sequence of n strings of parentheses ‘`()`’ and ‘`[]`’, one string a line

Output

A sequence of ‘`Yes`’ or ‘`No`’ on the output file

Sample Input	Sample Output
3 ([]) (([()]))) ([()[]()])()	Yes No Yes

Problem G

All in All

Time Limit: 1.000 second Memory Limit: 65536 KB

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*.

输入两个字符串*s*和*t*，判断是否可以从*t*中删除0个或者多个字符（其他字符顺序不变），得到字符串*s*。例如，abcde可以得到bce，但无法得到cb。

Input

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

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

Sample Input	Sample Output
`sequence subsequence`	`Yes`
`person compression`	`No`
`VERDI vivaVittorioEmanueleReDiItalia`	`Yes`
`caseDoesMatter CaseDoesMatter`	`No`

Problem H

加密专家

Time Limit: 1.000 second Memory Limit: 65536 KB

Ancient Roman empire had a strong government system with various departments, including a secret service department. The most popular ciphers were so called ***substitution cipher*** and ***permutation cipher***.
Substitution cipher changes all occurrences of each letter to some other letter. Substitutes for all letters must be different. For some letters substitute letter may coincide with the original letter. For example, applying substitution cipher that changes all letters from ‘**A**’ to ‘**Y**’ to the next ones in the alphabet, and changes ‘**Z**’ to ‘**A**’, to the message “**VICTORIOUS**” one gets the message “**WJDUPSJPVT**”.
Permutation cipher applies some permutation to the letters of the message. For example, applying the permutation ⟨2, 1, 5, 4, 3, 7, 6, 10, 9, 8⟩ to the message “**VICTORIOUS**” one gets the message “**IVOTCIRSUO**”.
It was quickly noticed that being applied separately, both substitution cipher and permutation cipher were rather weak. But when being combined, they were strong enough for those times. Thus, the most important messages were first encrypted using substitution cipher, and then the result was encrypted using permutation cipher. Encrypting the message “**VICTORIOUS**” with the combination of the ciphers described above one gets the message “**JWPUDJSTVP**”.
Archeologists have recently found the message engraved on a stone plate. At the first glance it seemed completely meaningless, so it was suggested that the message was encrypted with some substitution and permutation ciphers. They have conjectured the possible text of the original message that was encrypted, and now they want to check their conjecture. They need a computer program to do it, so you have to write one.

Input

Input file contains several test cases. Each of them consists of two lines. The first line contains the message engraved on the plate. Before encrypting, all spaces and punctuation marks were removed, so the encrypted message contains **only capital letters** of the English alphabet. The second line contains the original message that is conjectured to be encrypted in the message on the first line. It also contains **only capital letters** of the English alphabet.
The lengths of both lines of the input file are equal and do not exceed 100.

Output

For each test case, print one output line. Output ‘`YES`’ if the message on the first line of the input file could be the result of encrypting the message on the second line, or ‘`NO`’ in the other case.

Sample Input	Sample Output
JWPUDJSTVP VICTORIOUS MAMA ROME HAHA HEHE AAA AAA NEERCISTHEBEST SECRETMESSAGES	YES NO YES YES NO

Problem I

方块染色

Time Limit: 1.000 second

We have a machine for painting cubes. It is supplied with three different colors: blue, red and green. Each face of the cube gets one of these colors. The cube's faces are numbered as in Figure 1.

Since a cube has 6 faces, our machine can paint a face-numbered cube in $3^6 = 729$ different ways. When ignoring the face-numbers, the number of different paintings is much less, because a cube can be rotated. See example below.

We denote a painted cube by a string of 6 characters, where each character is a '**b**', '**r**', or '**g**'. The *i*-th character ($1 \leqslant i \leqslant 6$) from the left gives the color of face *i*. For example, Figure 2 is a picture of "**rbgggr**" and Figure 3 corresponds to "**rggbgr**". Notice that both cubes are painted in the same way: by rotating it around the vertical axis by 90°, the one changes into the other.

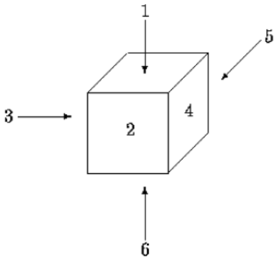

Figure 1

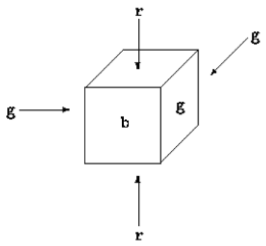

Figure 2

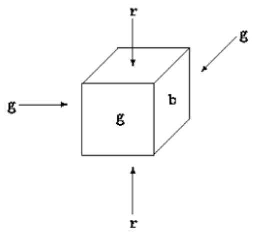

Figure 3

Input

The input of your program is a text file that ends with the standard end-of-file marker. Each line is a string of 12 characters. The first 6 characters of this string are the representation of a painted cube, the remaining 6 characters give you the representation of another cube. Your program determines whether these two cubes are painted in the same way, that is, whether by any combination of rotations one can be turned into the other. (Reflections are not allowed.)

Output

The output is a file of boolean. For each line of input, output contains '`TRUE`' if the second half can be obtained from the first half by rotation as describes above, '`FALSE`' otherwise.

Sample Input	Sample Output
rbgggrrggbgr rrrbbbrrbbbr rbgrbgrrrrrg	TRUE FALSE FALSE

Problem J

最后谁赢了？

Time Limit: 1.000 second   Memory Limit: 10000 KB

有两堆石子，数量任意，可以不同。游戏开始由两个人轮流取石子。游戏规定，每次有两种不同的取法，一是可以在任意的一堆中取走任意多的石子；二是可以在两堆中同时取走相同数量的石子。最后把石子全部取完者为胜者。现在给出初始的两堆石子的数目，如果轮到你先取，假设双方都采取最好的策略，问最后你是胜者还是败者。

Input

输入包含若干行，表示若干种石子的初始情况，其中每一行包含两个非负整数 a 和 b，表示两堆石子的数目，a 和 b 都不大于 1,000,000,000。

Output

输出对应也有若干行，每行包含一个数字 1 或 0，如果最后你是胜者，则为 1，反之，则为 0。

Sample Input	Sample Output

2 1
8 4
4 7
	0
	1
	0

Problem K

等式求解

Time Limit: 1.000 second   Memory Limit: 65536 KB

In this problem, you are to solve a very easy linear equation with only one variable x with no parentheses! An example of such equations is like the following:

$$2x - 4 + 5x + 300 = 98x$$

An expression in its general form, will contain a '=' character with two expressions on its sides. Each expression is made up of one or more terms combined by '+' or '-' operators. No unary plus or minus operators are allowed in the expressions. Each term is either a single integer, or an integer followed by the lower-case character x or the single character x which is equivalent to 1x.

You are to write a program to find the value of x that satisfies the equation. Note that it is possible for the equation to have no solution or have infinitely many. Your program must detect these cases too.

Input

The first number in the input line, t (1 <= t <= 10) is the number of test cases, followed by t lines of length at most 260 each containing an equation. There is no blank character in the equations and the variable is always represented by the lower-case character x. The coefficients are integers in the range (0...1000) inclusive.

Output

The output contains one line per test case containing the solution of the equation. If s is the solution to the equation, the output line should contain [s] (the floor of s, i.e., the largest integer number less than or equal to s). The output should be IMPOSSIBLE or IDENTITY if the equation has no solution or has infinite solutions, respectively. Note that the output is case-sensitive.

Sample Input	Sample Output
2 2x-4+5x+300=98x x+2=2+x	3 IDENTITY

Problem L

请勿“插队”

Time Limit: 3.000 second

有 t 个团队的人正在排一个长队。每次新来一个人时，如果他有队友在排队，那么这个新人会插队到最后一个队友身后。如果没有任何一个队友排队，则他会排到长队的队尾。输入每个团队中所有队员的编号，要求支持如下 3 种指令（前两种指令可以穿插进行）。

ENQUEUE：编号为 X 的人进入长队。

DEQUEUE：长队队首出队。

STOP：停止模拟。

对于每个 DEQUEUE 指令，输出出队的人的编号。

Input

输入文件中有一组或多组测试数据。每组测试数据开始有 t 个团队。下面 t 行，每行的第一个数字代表这个团队人数，后面是这几个人的编号。编号为 0 到 999999 之间的一个整数。每组测试数据以“STOP”结束。输入以 t==0 时结束。

温馨提示：一个测试用例可能包含最多 200000 个命令，所以注意你实现的队列应当是高效的。

Output

对于每组测试数据，先打印一句"Scenario #k"，k 是第几组数据。对于每一个"DEQUEUE"指令，输出一个出队的人的编号。每组测试数据后要换行，即使是最后一组测试数据。

Sample Input	Sample Output
2	Scenario #1
3 101 102 103	101
3 201 202 203	102
ENQUEUE 101	103
ENQUEUE 201	201
ENQUEUE 102	202
ENQUEUE 202	203
ENQUEUE 103	
ENQUEUE 203	Scenario #2
DEQUEUE	259001
DEQUEUE	259002

DEQUEUE DEQUEUE DEQUEUE DEQUEUE STOP 2 5 259001 259002 259003 259004 259005 6 260001 260002 260003 260004 260005 260006 ENQUEUE 259001 ENQUEUE 260001 ENQUEUE 259002 ENQUEUE 259003 ENQUEUE 259004 ENQUEUE 259005 DEQUEUE DEQUEUE ENQUEUE 260002 ENQUEUE 260003 DEQUEUE DEQUEUE DEQUEUE DEQUEUE STOP 0	259003 259004 259005 260001

Problem M

Perfect Server

Time Limit: 3.000 second

A network is composed of N computers connected by $N - 1$ communication links such that any two computers can be communicated via a unique route. Two computers are said to be ***adjacent*** if there is a communication link between them. The ***neighbors*** of a computer are the set of computers which are adjacent to it. In order to quickly access and retrieve large amounts of information, we need to select some computers acting as ***servers*** to provide resources to their neighbors. Note that a server can serve all its neighbors. A set of servers in the network forms a ***perfect service*** if every client (non-server) is served by **exactly one** server. The problem is to find a minimum number of servers which forms a perfect service, and we call this number ***perfect service number***.

We assume that N (≤ 10000) is a positive integer and these N computers are numbered from 1 to N. For example, Figure 1 illustrates a network comprised of six computers, where black nodes represent servers and white nodes represent clients. In Figure 1(a), servers 3 and 5 do not form a perfect service because client 4 is adjacent to both servers 3 and 5 and thus it is served by two servers which contradicts the assumption. Conversely, servers 3 and 4 form a perfect service as shown in Figure 1(b). This set also has the minimum cardinality. Therefore, the perfect service number of this example equals two.

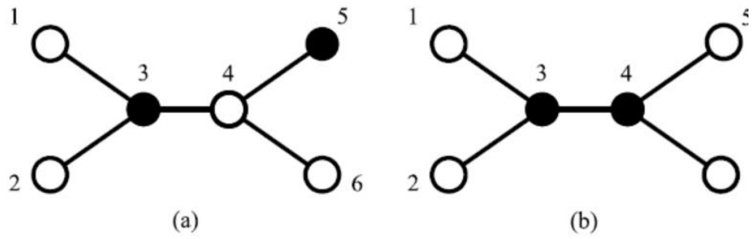

Figure 1

Your task is to write a program to compute the perfect service number.

Input

The input consists of a number of test cases. The format of each test case is as follows: The first line contains one positive integer, N, which represents the number of computers in the network. The next $N-1$ lines contain all of the communication links and one line for each link. Each line is represented by two positive integers separated by a single space. Finally, a '0' at the $(N + 1)$-th line indicates the end of the first test case.

The next test case starts after the previous ending symbol '0'. A '-1' indicates the end of the whole inputs.

Output

The output contains one line for each test case. Each line contains a positive integer, which is the perfect service number.

Sample Input	Sample Output

6
1 3
2 3
3 4
4 5
4 6
0
2
1 2
-1
	2
	1

Problem N

是谁在复读?

Time Limit: 1.000 second   Memory Limit: 65536 KB

The repetition number of a string is defined as the maximum number R such that the string can be partitioned into R same consecutive substrings. For example, the repetition number of "ababab" is 3 and "ababa" is 1.

Given a string containing lowercase letters, you are to find a substring of it with maximum repetition number.

Input

The input consists of multiple test cases. Each test case contains exactly one line, which gives a non-empty string consisting of lowercase letters. The length of the string will not be greater than 100,000.

The last test case is followed by a line containing a '#'.

Output

For each test case, print a line containing the test case number (beginning with 1) followed by the substring of maximum repetition number. If there are multiple substrings of maximum repetition number, print the lexicographically smallest one.

Sample Input	Sample Output
ccabababc daabbccaa #	Case 1: ababab Case 2: aa