Problem 1: Assessing Arithmetic Difficulty

The difficulty of multi-digit number additions can be determined by the number of carry operations of the addition problem. Write a program that will help school teachers assess the difficulty level of addition problems. Your program will display the number of carry operations for problems in a file.

Input

Each line contains two unsigned integers less than 10 digits. The last line contains two 0 values.

Output

For each line of input compute the number of carry operations as shown below.

Sample Input

123 456
123 557
123 577
666 666
7777 7777
99 7
999 9777
0 0

Sample Output

No carry operations.
1 carry operation.
2 carry operations.
3 carry operations.
4 carry operations.
2 carry operations.
4 carry operations.
Problem 2: LCD Display

Write a program that prints numbers the old fashion way using an LCD style pattern.

Input

The input file contains several lines, one line for each number to display. Each line contains two integers:

- the display size, \( s \) (1 \( \leq \) \( s \) \( \leq \) 10)
- the number to display, \( n \) (0 \( \leq \) \( n \) \( \leq \) 99,999,999)

The last line contains two 0 values.

Output

Print the numbers using \( s \) ‘-‘ for the horizontal segments and \( s \) ‘|’ for the vertical segments. Each digit occupies \( s + 2 \) columns and \( 2s + 3 \) rows. Fill all the white space occupied by the digits with blanks. There must be one column of blanks between two digits. Output a blank line after each number. See the examples below.

Sample Input

1 12345
1 67890
2 1234567890
3 67890
0 0

Sample Output

```
| | | | | |
| | | | | |
| | | | | |

| | | | | |
| | | | | |
| | | | | |

| | | | | |
| | | | | |
| | | | | |

| | | | | |
| | | | | |
| | | | | |

| | | | | |
| | | | | |
| | | | | |
```


Problem 3: A Tower of Turtles

Stacking turtles can be a useful way to see great sights. If it is not done properly though, it can lead to aches, cracking shells, and be susceptible to burping. Given a turtle to sit on the top, and a set of turtles to form the stack, determine the maximum safe height of the turtle stack.

Input

The input will consist of a number of scenarios. Each scenario can contain information for up to 5,608 turtles. The first turtle is the turtle that will sit on the top. Each line contains two integers, the weight and the strength of the turtle given in grams. The strength is the turtle’s carrying capacity, including its own weight. For example, a turtle with a weight of 170 g and a strength of 1,000 g can carry 830 g of turtles on its back. The end of a scenario is indicated with two 0s.

Output

Your output is an integer indicating the maximum number of turtles we can stack to build the tower without exceeding the strength of any turtle.

Sample Input

```
100    200
100    300
100    400
100    500
100    600
0       0
100    200
440    870
960    1730
250    430
640    750
700    1640
0       0
```

Sample Output

```
5
3
```
Problem 4: Smart AutoCorrect

Sometimes there’s nothing more frustrating than typing. Typing is a skill that can be improved the more you practice, but we programmers are lazy by nature. Whether it’s hitting the wrong key during an online game, or constantly messing up that one piece of coding, you wish there was an easy way to get better at typing. Well, simply put, there is.

A common typing error is to place your hands on the keyboard one row to the right of the correct position. Then “Q” is typed as “W” and “J” is typed as “K” and so on. Your task is to decode a message typed in this manner.

**Input**

- Input consists of several lines of text.
- Each line may contain digits, spaces, uppercase letters (except “Q”, “A”, “Z”), or punctuation shown above [except back-quote (‘)].
- Keys labeled with words [Tab, BackSp, Control, etc.] are not represented in the input.

**Output**

You are to replace each letter or punctuation symbol by the one immediately to its left on the QWERTY keyboard shown above. Spaces in the input should be echoed in the output.

<table>
<thead>
<tr>
<th>Example Input</th>
<th>Example Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>O JP[R UPI EPI;F GOC YJOD. HPPF ;IVL/ FP MPY GPTHRY YP YITM OY OM/ UPI NRRYRT YRDY YJOD 1234567890-= WERTYUIOP[]\ SDFGHJKL;‘ XCVBNM,./</td>
<td>I HOPE YOU WOULD FIX THIS, GOOD LUCK. DO NOT FORGET TO TURN IT IN. YOU BETTER TEST THIS `1234567890- QWERTYUIOP[] ASDFGHJKL; ZXCVBNM,.</td>
</tr>
</tbody>
</table>
Problem 5: Arranged Marriage

The Waka-Waka Island is a wonderful tropical paradise, but for the past 40 years there has been only one TV station and no Internet. In an attempt to control the rise in population, the prime minister has asked you to come up with a plan to solve this problem. After few hours of watching TV (reruns), you came up with the brilliant idea of targeting marriages and decided to follow this plan:

A man with a name M is allowed to marry a woman with name W, only if M is a subsequence of W or W is a subsequence of M.

In general, X is said to be a subsequence of Y, if X can be obtained by deleting some elements of Y without changing the order of the remaining elements.

Your task is to determine whether a couple is allowed to marry or not, according to this rule.

Input

1. The first line contains an integer T, the number of test cases (1 ≤ T ≤ 100).
2. T test cases follow. Each test case (line) contains two space separated strings M and W.
3. 1 ≤ |M|, |W| ≤ 255 (|A| denotes the length of the string A)
4. All names consist of lowercase English letters only.

Output

1. For each test case print "YES" if they are allowed to marry, else print "NO".

<table>
<thead>
<tr>
<th>Example Input</th>
<th>Example Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td></td>
</tr>
<tr>
<td>a alex</td>
<td>YES</td>
</tr>
<tr>
<td>tony o</td>
<td>YES</td>
</tr>
<tr>
<td>jenny jenny</td>
<td>YES</td>
</tr>
<tr>
<td>justin mary</td>
<td>NO</td>
</tr>
<tr>
<td>pat patricia</td>
<td>YES</td>
</tr>
<tr>
<td>santana ana</td>
<td>YES</td>
</tr>
<tr>
<td>bob roberta</td>
<td>NO</td>
</tr>
<tr>
<td>john johanna</td>
<td>YES</td>
</tr>
<tr>
<td>robert berta</td>
<td>NO</td>
</tr>
<tr>
<td>luis luisa</td>
<td>YES</td>
</tr>
<tr>
<td>x mandy</td>
<td>NO</td>
</tr>
<tr>
<td>antonio ann</td>
<td>YES</td>
</tr>
<tr>
<td>hakuna matata</td>
<td>NO</td>
</tr>
<tr>
<td>dan dillian</td>
<td>YES</td>
</tr>
<tr>
<td>pumbaa uma</td>
<td>YES</td>
</tr>
</tbody>
</table>
Problem 6: Electric Cars

The availability of charging stations is a factor that needs to be considered when planning a trip in an electric car. In this problem, compute the minimum trip distance between two points.

A graph which represents a road network will be provided, you may assume that a charging station is at each node. What is the length shortest path between two nodes, taking into account the range restriction of the car.

Input

1. A non negative integer describing the range of the car.
2. The program should exit if the range is zero.
3. A comma separated list of the nodes. Each node is described by a single upper case letter.
4. The next line has a positive integer describing how many edges are in the graph.
5. Each edge is described by a comma separated list of the two vertices, and the distance. Edges are bi-directional.
6. After the edges, comma-separated pairs of edges denote potential trips.
7. Queries about this graph will be terminated by a period (’.’) on a line by itself.
8. The program should continue accepting graphs and queries until the range of the car is 0.

Output

1. The program should output the shortest distance of the path that a car with that range can take to get between the two points.
2. If there is no path possible for that car, then the program should output -1.

Sample Input

```plaintext
50
A,B,C
3
A,B,75
C,A,25
C,B,30
A,B
.
25
A,B,C,D
4
A,B,15
C,A,35
C,B,30
D,B,20
A,B
A,C
D,A
.
0
```

Sample Output

```plaintext
55
15
-1
35
```
Problem 7: Emirp Numbers

An emirp number is a prime number where if you reverse the digits the number is a different prime number. Given a long list of numbers, indicate which ones are emirp numbers and which ones are not.

Input

1. The list for testing will be long. Your program must complete within a few seconds.
2. The numbers tested will have no more than 6 digits.
3. Each number to be tested will be on a separate line.

Output

1. Output a 1 if it is an emirp number and a 0 if it is not.

Sample Input

```
5
7
11
13
15
17
```

Sample Output

```
0
0
0
1
0
1
```
Problem 8: Condorcet Voting

Rather than voting for a single candidate, some voting methods allow voters to rank their preferences. A Condorcet winner of an election is a candidate that wins against any other candidate when doing a pairwise comparison between the two. There can be at most one condorcet winner of an election, and it is possible there is no winner. Given a set of candidates and a set of votes determine who the condorcet winner is, if any. Your program should keep evaluating different test cases until a period ('.') is given for the list of candidates.

Input

1. The first line will be a comma separated list of candidates. Each candidate will be represented by a single upper case letter.

2. Each vote will be on a separate line. The vote will consist of a comma separated list of the candidates.

3. Each vote will provide an ordering of candidates based on their position. The candidate on the left is preferred to the candidate on the right.

4. Candidates not appearing in a vote are equally unfavored.

Output

1. Output the condorcet winner of the election.

2. If there is no winner, output a 0.

Sample Input

```
A,B,C
A,B
B,A,C
C,A,B
.
A,B,C
A,B
B,C
C,A
.
A,B,C
C,A,B
B,A,C
B,C
.
A,B,C
C,A,B
B,A,C
C,A,B
B,A,C
.
```

Sample Output

```
A
0
B
0
```