ACM-ICPC Texas State – Mexico
Invitational Programming Contest
Mar. 31, 2018

Problem 1: Currency

Description

When you travel to Hong Kong, the first thing is currency exchange because foreign currencies are not accepted in Hong Kong. Given the exchange rate of different currencies and the amount of Hong Kong Dollars you need, please calculate the corresponding amount of foreign currencies.

Input Specification

The input contains N+2 lines. In each test case, the first line contains a number N (N <= 15), which means the number of foreign currencies. In the following N lines, each line contains a string and a real number, which indicate the name of the foreign currency and the exchange rate (100 units of the foreign currency against HKD). The last line contains a real number, which shows the amount of HKD you want to spend in Hong Kong.

Output Specification

The output should contain N lines. Each line contains the name of foreign currency and the amount you need to have in that currency (in order to exchange the HKD you want to spend). Real numbers in the output should be rounded to 0.01. The output order of currencies should be the same as the input one.

Sample Input and Output

Input	Output
8 CNY 112.42 USD 776.22 GBP 973.50 AUD 589.95 CAD 587.70 EUR 826.20 CHF 774.05 DKK 111.10 1500.00	CNY 1334.28 USD 193.24 GBP 154.08 AUD 254.26 CAD 255.23 EUR 181.55 CHF 193.79 DKK 1350.14

Problem 2: Anagram Groups

World-renowned Prof. A. N. Agram's current research deals with large anagram groups. He has just found a new application for his theory on the distribution of characters in English language texts. Given such a text, you are to find the largest anagram groups.

A text is a sequence of words. A word *w* is an anagram of a word *v* if and only if there is some permutation *p* of character positions that takes *w* to *v*. Then, *w* and *v* are in the same anagram group. The size of an anagram group is the number of words in that group. Find the 5 largest anagram groups.

Input Specification

The input contains words composed of lowercase alphabetic characters (one word in each line). It is terminated by EOF.

Output Specification

Output the 5 largest anagram groups. If there are less than 5 groups, output them all. Sort the groups by decreasing size. Break ties lexicographically by the lexicographical smallest element. For each group output, print its size and its member words (separate each word using a space). Sort the member words lexicographically and print equal words only once.

Sample Input and Output

Input	Output
undisplayed trace tea singleton eta eat displayed crate cater carte caret beta beat bate ate abet	Group of size 5: caret carte cater crate trace . Group of size 4: abet bate beat beta . Group of size 4: ate eat eta tea . Group of size 1: displayed . Group of size 1: singleton .

Problem 3: Let it Bead

Description

"Let it Bead" company is located upstairs at 700 Cannery Row in Monterey, CA. As you can deduce from the company name, their business is beads. Their PR department found out that customers are interested in buying colored bracelets. However, over 90 percent of the target audience insists that the bracelets be unique. (Just imagine what happened if two women showed up at the same party wearing identical bracelets!) It's a good thing that bracelets can have different lengths and need not be made of beads of one color. Help the boss estimating maximum profit by calculating how many different bracelets can be produced.

A bracelet is a ring-like sequence of *s* beads each of which can have one of *c* distinct colors. The ring is closed, i.e. has no beginning or end, and has no direction. Assume an unlimited supply of beads of each color. For different values of *s* and *c*, calculate the number of different bracelets that can be made.

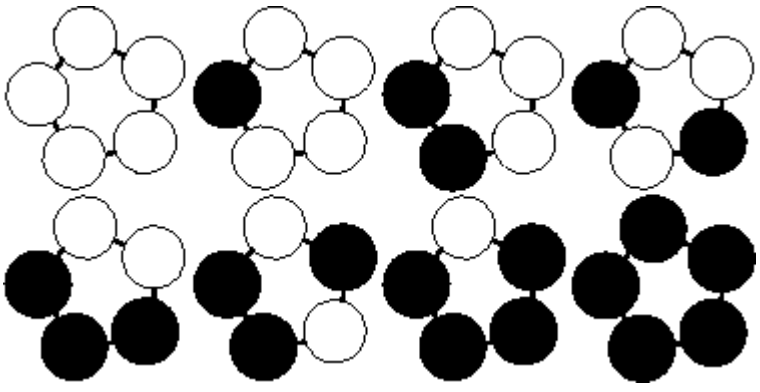
Input Specification

Every line of the input file defines a test case and contains two integers: the number of available colors *c* followed by the length of the bracelets *s*. Input is terminated by *c=s=0*. Otherwise, both are positive, and, due to technical difficulties in the bracelet-fabrication-machine, *cs<=32*, i.e. their product does not exceed 32.

Output Specification

For each test case output on a single line the number of unique bracelets. The figure below shows the 8 different bracelets that can be made with 2 colors and 5 beads.

Sample Input	Sample Output
1 1	1
2 1	2
2 2	3
5 1	5
2 5	8
2 6	13
6 2	21
0 0	



Problem 4: Valid Parentheses

Description

Given a string containing just the characters '(' and ')' determine if the input string is valid.

The brackets must close in the correct order, "()", "()()" and "(())" are all valid but "(" and ")(" are not.

Input Specification

The input begins with the number t of test cases in a single line (t<=10). In each of the next t lines there is a non-empty parentheses string.

Output Specification

For every test case print "true" or "false" indicates whether the input string is valid.

Sample Input and Output

Input	Output
5	true
()	true
()()	false
(())	false
(	false
)(	

Problem 5: Golden Leaves

Description

Jack and Bob are two kids who love prime numbers. Jack just received his birthday gift from his father: *N* golden leaves of his favorite Ginkgo tree. Since Jack and Bob are really good friends, Jack would like to share these golden leaves with Bob. In order to make Bob happy, Jack plans to give *p* pieces to Bob where *p* is a prime. Similarly, *N*-*p* must be a prime, too. Otherwise, Jack will be unhappy. If there are multiple feasible values for the prime number *p*, then Jack will choose the smallest one so that he can keep as many as his favorite golden leaves.

Write a program to check whether such a number *p* exists. If *p* does exist, calculate how many golden leaves Jack can keep.

Input Specification

The input contains at most 2,500 test cases. Each test case contains a positive integer *N* (*N* is at most 1,000,000) in a line. A single 0 indicates the end of input.

Output Specification

For each case, print -1 if *p* does not exists. Otherwise, print the maximum number of golden leaves that Jack can keep.

Sample Input

```
123
456
789
0
```

Sample Output

```
-1
449
787
```

Problem 6: All in All

Description

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 test cases. 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