Revision

Data and File Structures Laboratory

http://www.isical.ac.in/~dfslab/2019/index.html

Problems – I

1. An anagram is a word or phrase formed by rearranging the letters of another word or phrase. For example, *carthorse* is an anagram of *orchestra*. Blanks within a phrase are ignored when forming anagrams. Thus, *orchestra* and *horse cart* are also anagrams. Write a program that reads a list of phrases and prints all pairs of anagrams occurring in the list.

   Input: The input will consist of upto 100 lines. A completely empty or blank line signals the end of input. Each line constitutes one phrase.

   Output: Some number of lines, each line containing two anagrammatic phrases separated by = . Each anagram pair should be printed exactly once; the phrase occurring earlier in the input list should appear on the left. If there are no anagrams in the input, your program should not produce any output.

Problems – II

Example input:

```
carthorse
horse
horse cart
i do not know u
ok i now donut
orchestra
```

Example output:

```
carthorse = horse cart
carthorse = orchestra
horse cart = orchestra
i do not know u = ok i now donut
```

SOURCE: Old Dominion University programming contest

Problems – III

2. A character string is said to have period k if it can be formed by concatenating one or more repetitions of another string of length k. For example, the string `abcabcabcabc` has period 3, since it can be formed by 4 repetitions of the string "`abc`. It also has periods 6 (two repetitions of `abcabc`) and 12 (one repetition of `abcabcabcabc`). Write a program to read a character string and determine its smallest period. You should ignore case for this problem.

   Input: A single character string of up to 80 non-blank characters.

   Output: An integer denoting the smallest period of the input string.

   Example input: `hahaha`

   Example output: 2

   SOURCE: Old Dominion University programming contest

Problems – IV

3. Write a program that plays an interactive game of dots and boxes with the user. You may refer to the following sites for more details about the game:

   - `https://en.wikipedia.org/wiki/Dots_and_Boxes`
   - `http://dotsandboxes.org/`
   - `https://www.mathsisfun.com/games/dots-and-boxes.html`
   - `http://www.math.ucla.edu/~tom/Games/dots&boxes.html`

   Your program should take two command line arguments that specify the number of rows and columns in the game grid.

Problems – V

4. We define the *super digit* of an integer n as follows:

    - if n has only 1 digit, then its super digit is n;
    - otherwise, the super digit of n is the super digit of the number formed by adding together all the digits of n.

    Example: super digit of 9875 = 2.

    Write a program that outputs the superdigit of a given integer.

    SOURCE: Hackerrank

5. Georg Cantor proved that the set of rational numbers is enumerable, by constructing an explicit enumeration of rational numbers as follows.

    ```
    1/1 1/2 1/3 1/4 1/5 ...
    2/1 2/2 2/3 2/4 ...
    3/1 3/2 3/3 ...
    4/1 4/2 ...
    5/1 ...
    ```

Problems – VI

In the above diagram, the first term is `1/1`, the second term is `1/2`, the third term is `2/1`, the fourth term is `3/1`, the fifth term is `2/2`, and so on. Write a program that, given a natural number i, finds the i-th rational number in Cantor's enumeration.

Input: A single positive integer n specifying the number of test cases, followed by n more positive integers.

Output: For each test case, print the corresponding rational number in Cantor's enumeration.

Example input:

```
3
3
14
7
```

Example output:

```
2/1
2/4
1/4
```