**Loops**:

A sequence consists of integer numbers and ends with the number 0. Determine how many elements of this sequence are greater than their neighbors above.

: *Using for loop,and if -else error index out of range error was thrown.*

*So used try--except*

Determine the number of even elements in the sequence ending with the number 0.

: *Using for loop,and if -else error index out of range error was thrown.*

*So used try--except*

For given integer n ≤ 9 print a ladder of n steps. The k-th step consists of the integers from 1 to k without spaces between them.To do that, you can use the sep and end arguments for the function print()

**List problems:**

Given a list of numbers, determine and print the quantity of elements that are greater than both of their neighbors.The first and the last items of the list shouldn't be considered because they don't have two neighbors.

Given a list of numbers, find and print all the elements that are greater than the previous element.

Given a list of numbers, swap adjacent items in pairs (A[0] with A[1], A[2] with A[3], etc.). Print the resulting list. If a list has an odd number of elements, leave the last element in place.

*Even if it was odd, no extra check is done.*

Given a list of unique numbers, swap the minimal and maximal elements of this list. Print the resulting list.

Given a list of numbers, count how many element pairs have the same value (are equal). Any two elements that are equal to each other should be counted exactly once.

--didn’t do

Given a list of numbers, find and print the elements that appear in the list only once. The elements must be printed in the order in which they occur in the original list. –dint do

In chess it is known that it is possible to place 8 queens on an 8×8 chess board such that none of them can attack another. Given a placement of 8 queens on the board, determine if there is a pair of queens that can attach each other on the next move. Print the word NO if no queen can attack another, otherwise print YES. The input consists of eight coordinate pairs, one pair per line, with each pair giving the position of a queen on a standard chess board with rows and columns numbered starting at 1.

**Function/Recursion:**

Write two functions that takes as input a number n and returns the factorial of that number. One function should use loops and other use recursion to solve the same.

**Some generic problems that can solved using strings, lists, loops, function and/or recursion:**

Problem J2: Happy or Sad Problem Description We often include emoticons in our text messages to indicate how we are feeling. The three consecutive characters :-) indicate a happy face and the three consecutive characters :-( indicate a sad face.

Write a program to determine the overall mood of a message.

Input Specification There will be one line of input that contains between 1 and 255 characters. Output Specification The output is determined by the following rules:

• If the input line does not contain any happy or sad emoticons, output none.

• Otherwise, if the input line contains an equal number of happy and sad emoticons, output unsure.

• Otherwise, if the input line contains more happy than sad emoticons, output happy.

• Otherwise, if the input line contains more sad than happy emoticons, output sad.

Sample Input 1 How are you :-) doing :-( today :-)?

Output for Sample Input 1 happy

Sample Input 2 :)

Output for Sample Input 2 none

Sample Input 3 This:-(is str:-(:-(ange te:-)xt.

Problem J3: Problem Description In Sweden, there is a simple child’s game similar to Pig Latin called Rovarspr ¨ aket (Robbers Lan- ˚ guage).,

every consonant is replaced by three letters, in the following ˚ order:

• the consonant itself;

• the vowel closest to the consonant in the alphabet (e.g., if the consonant is d, then the closest vowel is e), with the rule that if the consonant falls exactly between two vowels, then the vowel closer to the start of the alphabet will be chosen (e.g., if the consonant is c, then the closest vowel is a);

• the next consonant in the alphabet following the original consonant (e.g., if the consonant is d, then the next consonant is f) except if the original consonant is z, in which case the next consonant is z as well. Vowels in the word remain the same. (Vowels are a, e, i, o, u and all other letters are consonants.)

Write a program that translates a word from English into Rovarspr .

Input Specification The input consists of one word entirely composed of lower-case letters. There will be at least one letter and no more than 30 letters in this word.

Output Specification

Output the word as it would be translated into Rovarspr on one line.

Sample Input 1 joy

Output for Sample Input 1 jikoyuz

Sample Input 2 ham

Output for Sample Input 2 hijamon

Problem J4: Wait Time Problem Description You exchange text messages with your friends. Since you receive so many messages, you want to measure how long your friends have to wait for your replies. Your message device records each received and sent message in order using the following two kinds of entries:

• R X indicates a message was received from a friend numbered X;

• S X indicates a message was sent to a friend numbered X.

Your message device sends and receives messages instantaneously, and for each consecutive pair of entries described above, either

• a single entry W X is recorded in between them indicating they occur X seconds apart, or

• there is no entry between them and they occur one second apart.

Several rules of message etiquette are always followed:

• the only messages you send are replies to messages that you have received;

• you send at most one reply to any message from any particular friend;

• your friends do not send a subsequent message until you have replied to their previous message.

The wait time for a message is the time that passes between when you receive it and the time you reply to it. If a friend X received a reply to each message they sent, the total wait time for friend X is the sum of all wait times for all messages from friend X. Otherwise, the total wait time for friend X is −1. Your job is to determine the total wait time for each friend.

Input Specification The input consists of the integer M (1 ≤ M ≤ 20), followed by M lines, where each line consists of one character (W, R, or S), followed by a space, followed by an integer X (1 ≤ X ≤ 100). These M lines are the entries described above (in order).

Output Specification Output one line for each friend that sent a message in the form X T where X is a friend number and T is the total wait time for friend X. The lines are in increasing order of the friend numbers