National Institute of Technology, Calicut
Department of Computer Science & Engineering

Object Oriented Systems Lab

EVALUATION 3 QUESTIONS_SET2

1. We can assign a value to each character in a word, based on their position in the alphabet (a = 1, b = 2, ... , z = 26).

A balanced word is one where the sum of values on the left-hand side of the word equals the sum of values on the right-hand side.
For odd length words, the middle character (balance point) is ignored.

Write a function that returns true if the word is balanced, and false if it's not.

Constrains:

(i) All words will be lowercase.
(ii) Should have minimum of 2 characters.

Note: Palindromic words will always be balanced.

Input format:
The first line, the user should enter the string

Output format:
Print whether it is balanced or not (i.e, print true or false)

Sample Input and Output

INPUT	***OUTPUT***
zips	True 26+9|16+19 = 35|35

INPUT	***OUTPUT***
brake	False 2+18|11+5 = 20|16

2. Musical instruments have a range of notes to play, some instruments having a much larger range than others.

Given the following ranges for the instrument, create a function that returns true if a given note is within a given instrument's range. Otherwise, return false.

INSTRUMENT	***RANGE***
Piccolo	D4-C7
Tuba	D1-F4
Guitar	E3-E6
Piano	A0-C8
Violin	G3-A7

Note:

(I) Tests will only include natural notes (i.e. you will only deal with letters and numbers, no special characters).

(II) The musical scale follows this pattern:
... A1, B1, C1, D1, E1, F1, G1, A2, B2 ...A8,B8,C8,E8,F8,G8

Input format:
<instrument_name><space><musical_scale>

Output format:
True or False (i.e whether the musical scale is in the range of the instrument or not.

Sample Input and Output

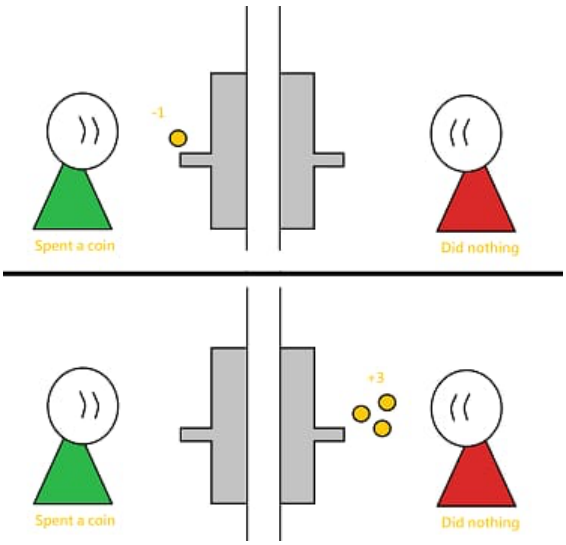
INPUT	***OUTPUT***
Piccolo A3 Violin G6 Piano C8	False True True

3. Let's say that there exists a machine that gives out free coins, but with a twist!

Separating two people is a wall, and this machine is placed in such a way that both people are able to access it. Spending a coin in this machine will give the person on the other side 3 coins and vice versa.

If both people continually spend coins for each other (SHARING), then they'll both gain a net profit of 2 coins per turn. However, there is always the possibility for someone to act selfishly (STEALING): they spend no coins, yet they still receive the generous 3 coin gift from the other person!

Here's an example of Red taking advantage of Green! Red chose to betray



The Challenge
Assuming that both people start with 3 coins each, create a function that calculates both people's final number of coins. You will be given two arrays of strings, with each string being the words "share" or "steal".

Input Format :
[The contribution of person 1 array],[The contribution of person 2 array]
Please note: if a person is contributing, it should be given as “share” and if a person is not contributing, it should be given as “steal”.

Please note:

for every test case, initially, the #coins with person 1= #coins with person 2= 3
i.e, the previously gained coins are not considered when the user is inputting the next testcase.

Output format:

Return the array list of 2 elements, where the first element corresponds to the #coins finally with person1 and second element corresponds to the #coins finally available with person2.

Sample Input and Output

INPUT	***OUTPUT***
([share], [share])	[5, 5]
([steal], [share])	[6, 2]
([steal], [steal])	[3, 3]
([share, share, share], [steal, share, steal])	[3, 11]

Explanation for the above sample outputs:
(["share"], ["share"]) ➞ [5, 5]
// Both people spend one coin, and receive 3 coins each.

(["steal"], ["share"]) ➞ [6, 2]
// Person 1 gains 3 coins, while person 2 loses a coin.

(["steal"], ["steal"]) ➞ [3, 3]
// Neither person spends any coins and so no one gets rewarded.

(["share", "share", "share"], ["steal", "share", "steal"]) ➞ [3, 11]

Note:
(i) No tests will include a negative number of coins.
(ii) All words will be given in lowercase.
(iii) This challenge is adapted from a famous game theory example called the Prisoner's Dilemma.

To Understand more about “The prisoner's dilemma”, read the following:

The prisoner's dilemma is a standard example of a game analyzed in game theory that shows why two completely rational individuals might not cooperate, even if it appears that it is in their best interests to do so. It was originally framed by Merrill Flood and Melvin Dresher while working at RAND in 1950. Albert W. Tucker formalized the game with prison sentence rewards and named it "prisoner's dilemma",[1] a version of which was stated by William Poundstone in his 1993 book Prisoner's Dilemma as:

Two members of a criminal gang are arrested and imprisoned. Each prisoner is in solitary confinement with no means of speaking to or exchanging messages with the other. The police admit they don't have enough evidence to convict the pair on the principal charge. They plan to sentence both to a year in prison on a lesser charge. Simultaneously, the police offer each prisoner a Faustian bargain.

The possible outcomes are:

If A and B each betray the other, each of them serves two years in prison
If A betrays B but B remains silent, A will be set free and B will serve three years in prison
If A remains silent but B betrays A, A will serve three years in prison and B will be set free
If A and B both remain silent, both of them will serve one year in prison (on the lesser charge).

It is implied that neither prisoner will have the opportunity to reward or punish their partner other than the prison sentences they get. It is also implied that the each prisoner's decision by itself will not affect their reputation in the future. Because betraying a partner offers a greater reward than cooperating with them, all purely rational self-interested prisoners will betray the other, meaning the only possible outcome for two purely rational prisoners is for them to betray each other, even though mutual cooperation would yield a greater reward.