**Exercise 1: 2020/2021**

The PoliTo company wants to create an application capable of tracking the sales of its products in order to identify any counterfeits. The application must analyze the contents of two files. A first file of text products.txt is created by the parent company and contains, for each product manufactured, the identifier of the unique one official dealer who is authorized to sell the product. There are two for each line in the file information (strings separated by a space):

product\_id reseller\_id

Each product and each seller are identified by a unique alphanumeric code.

A second text file purchases.txt contains information on sales that have been recorded by buyers. The file contains on each line the code of the product purchased and the retailer who managed the delivery.

Since a product can only be sold by the official dealer authorized by the parent company, any purchase of that product by a buyer through an not official retailer must be reported as a suspicion to then allow the parent company to carry out the necessary checks.

The parent company therefore asks you to write a Python program which, after reading the two input files, prints the list of possible suspicious sales. Specifically, for each product sold by one or more non-resellers authorized, the program must print the product code on output (in the format indicated in the example below) in question, the official reseller and the list of all resellers with which customers have registered the sale.

\*\*\* Note \*\*\*: there is no sorting of the two input files

**Example products.txt:**

P234HF22222 r1011

P234HF22223 r1112

P234HF22225 r1114

P111TG11115 r1015

P111TG11116 r1216

P331LS00110 r1017

P331LS00120 r1318

P331LS00130 r1019

**Example purchases.txt**

P234HF22223 r1112

P111TG11115 r1015

P111TG11115 r1216

P234HF22222 r1011

P331LS00110 r1014

P331LS00120 r1318

P331LS00130 r1019

P234HF22225 r1114

P234HF22223 r1114

**Program output**

Suspicious transactions list

Product code: P234HF22223

Official dealer: r1112

Non official dealers: r1114

Product code: P111TG11115

Official dealer: r1015

Non official dealers: r1216

Product code: P331LS00110

Official dealer: r1017

Non official dealers: r1014

**Exercise 2: 2020/2021**

Write a program for reading a text file composed of multiple lines and print all the triplets of adjacent words of the same length in uppercase. Punctuation marks should not be considered part of the words. The file does not contain genitives (possessive) nor contractions, such as n't, 've.

For example, given the file strawberry-short.txt:

No one I think is in my tree

I mean it must be high or low

That is you cannot, you know, tune in, but it is all right

That is I think it is not too bad...

Let me take you down, because I am going to

Strawberry Fields

Nothing is real, and nothing to get hung about

Strawberry Fields forever

**The output should be:**

('IS', 'IN', 'MY')

('NOT', 'TOO', 'BAD')

('TOO', 'BAD', 'LET')

Please note that too and bad appear in two different triplets. While for the complete lyrics (file strawberry.txt), the output should be:

('EASY', 'WITH', 'EYES')

('ALL', 'YOU', 'SEE')

('IS', 'IN', 'MY')

('NOT', 'TOO', 'BAD')

('TOO', 'BAD', 'LET')

('IT', 'IS', 'ME')

**Exercise 3: 2021/2022**

Alice has 42 magic boxes.

Each box has infinite capacity, but can only store one type of objects. The type itself is not relevant, but once an object has been inserted in a box, then only objects of that type can be added. For instance, after inserting an apple, any number of apples can be added, even tons, but not a single banana.

If a magic box is emptied, it can be refilled with a new type of objects. For instance, after removing all the apples from the box, a banana can be stored; and after that, any number of bananas, but not apples anymore.

Bob is handling Alice a sequence of objects; Alice should store them in the magic boxes according to previous notes. At the same time, Carl is asking Alice for objects; Alice should take the object Carl is asking for from one box, and handle it to him.

Write a program to simulate the behavior of Alice, Bob, and Carl. A text file named 'actions.txt' contains the actions performed by Bob and Carl, one per line, in the form "Bob gives a OBJECT" or "Carl takes a OBJECT". The program should check what happen, reporting a message if Alice is not able to respond correctly because either she can't store Bob's object in a box, or she can't give Carl the requested object because it's not available.

The file is correct. Object names are a single word and are written in CAPITAL.

For instance (and assuming that Alice has only 2 boxes):

Bob gives a APPLE

Bob gives a BANANA

Bob gives a APPLE

Bob gives a CHERRY

Generates an error, because Alice cannot store a CHERRY

Bob gives a APPLE

Bob gives a BANANA

Bob gives a APPLE

Carl takes a CHERRY

Bob gives a CHERRY

Generates an error, because Alice cannot give a CHERRY

Bob gives a APPLE

Bob gives a BANANA

Bob gives a APPLE

Carl takes a BANANA

Bob gives a CHERRY

Generates no errors, as the box with the first BANANA is emptied and can later store a CHERRY.