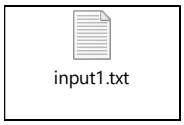
Assignment #4
Course: CPSC 442 (Python For Data Science)

Please Note: You need to use *try except* block wherever the need arises for any of the following questions. You would also need to *log the exceptions*.

1. Your task is to write a function to print valid phone numbers. You need to accept the inputs from the file ***(input1.txt)*** and output the valid phone numbers on the console.
   Format for valid phone numbers is:
   (xxx) xxx-xxxx
   xxx-xxx-xxxx
   where x would be a number.
   Sample Input:

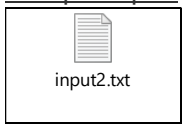



```
Example:
Input: 987-123-4567
       123 456 7890
       (123) 456-7890

Output: 987-123-4567
        (123) 456-7890
```

Hint: You may use regular expressions. (import re)

2. You are given a file as an input. Your task is to *transpose* the contents of the file. The input file may include uneven lines of whitespace which you need to handle. Note, the output matrix should not have whitespaces.
   Write a function to output the transposed content on the console as well as to an output file.
   Do not use numpy or pandas for this question.
   Sample Input:



Example:

```
Example 1:

Input: this is test

my name xyz


python programming class
```

Whitespaces

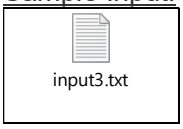
Whitespaces

```
Output: [('this', 'my', 'python'), ('is', 'name', 'programming'), ('test', 'xyz', 'cl
ass')]
```

3. Your task is to create a library manager as a menu-driven console application. The system should have the following functionalities in their menu:
   i. Add books
   ii. Remove books
   iii. Add membership
   iv. Remove membership
   v. Rent book
   vi. List books
   vii. List members
   viii. Exit

   You need to create separate files to store information of books, members, book rentals.
   When the system starts, it should import all contents of the file into the system, if the files are not empty.
   All-important fields should be checked for empty or incorrect format. Examples: name, DOB, etc. should not be blank. Phone numbers should be a standard format, etc.
   Also, you may add any other related functionality as well.

4. Complete the remaining tic-tac-toe game. Tic-tac-toe always has a square matrix. The game would be in easy-mode, i.e. 3x3 matrix. You would be given positions of the already half-played game in input file. Your task is to first fill in the matrix from the positions given in the file.
   Then, you need to play for the remaining positions. In the end, you need to output the ***possible winner(A or B) or Draw.***
   Sample Input:




```
Example 1:
Input:  1:X
        9:O
        5:X
        6:O
```

3:X

2:O

Output: Player 1 is A

Player 2 is B

Winner: A

Explanation: The input format is ***location:X or O***. The location in the matrix starts from 1. So, 1 corresponds to [0][0] of the matrix. Similarly, 6 corresponds to [1][2]. After filling the matrix:

X O X

X O

O

As per the inputs, the next player would be A and he would play an **X.** The next best position for him would be 7 which corresponds to [2][0] for a win.

X O X

X O

X O

So, here the output would be A.

Hint: You may use recursion technique for this question.