Python Inlab Exercises

Instructions

1. COPY THE DATA FOLDER INSIDE YOUR SOLUTION DIRECTORY
2. **Submission format:**

```
RollNumber_InLab3.tar.gz
|
+- RollNumber_InLab3
   |- Data/
   |- Q1.py
   |- Q2.py
   |- Q3.py
   |- Q4.py
   |- Q5.py
```

Q1

Write a program in python which will take n numbers as input (on stdin) and print on separate lines whether the number is square number or not. You need to implement a function isSquare() which prints whether a number is a square or not. **Don't use any library.**

Input format:
The first line contains n, the number of integers.
The next line contains the n integers.

Sample Input:

```
5
4 9 8 7 1
```

Note : Strictly follow the input format

Expected Output:

```
4 is a square number
9 is a square number
8 is not a square number
7 is not a square number
1 is a square number
```

Q2 Sherlocked

Sherlock has designed a security software in his favourite language Python (see Q2.py). It takes a pin (integer) ranging from 0 to 9999 as input and if it is correct the safe gets opened. If you get the pin wrong once, you get another chance. If you miss it again, the software goes into full alarm mode (prints "Full alarm").

Sample interaction:

```
Enter the pin: 1234
Wrong PIN! Try again
Enter the pin: 0911
Full alarm
```

His arch-nemesis, Jim Moriarty finds a flaw in his code and figures out if he gets the pin wrong at the first try, he can enter a string, which makes the program crash.

```
Enter the pin: 1234
Wrong PIN! Try again
Enter the pin: Sherlock is dumb
*Program crashes*
```

Then he starts the program again and tries a new pin, till he gets the correct one. Sherlock's assistant Dr. Watson, not being a brilliant programmer like yourself, asks you to add some changes to the code, so that, no matter what Jim Moriarty does, the program goes into Full alarm on any illegal input.

Hint: Dr. Watson mentions that Sherlock never studied the chapter for Exception Handling when he studied Python.
NOTE: You must edit `Q2.py` and submit it.

Clarifications :

1. You should output full alarm on illegal input on any trial (both 1st and 2nd) and exit the code.
2. 04949 should be converted to 4949 and then proceed further. So it should NOT give a FULL ALARM

Q3

Dr. Strange knows that Thanos is coming for the Time Stone. He knows that Iron Man, Black Widow, Hulk, Thor, Captain America, War Machine, Captain Marvel, Hawkeye etc. are the only hope to stop Thanos.
To save them Dr. Strange asks for your help. He gives you a file (`Heroes.txt`) with the name of the heroes he needs to save and a directory (`Avengers_Universe`) full of files. Some of the files inside the directory contain the names of these heroes.
You need to find those files and move them to a new directory (inside `Data/`) named `Survivors_of_Snap`.

Before Dr. Strange leaves for the fight, he says, *"Make good use of the os library"*.
The future of the Avengers Universe is in your hands, the coding avenger.

TL;DR

Given a directory of files, find the files which have a keyword present in them amongst a list of keywords. Create a new directory and move those files in that directory.

Assume that,

1. Each file will have only one keyword.
2. A keyword appears only once in the directory.
3. In each file, only one line will contain the keyword and **that line will contain no word** other than that keyword.

NOTE: Use relative paths, not absolute paths.

WRONG: Absolute path: `/home/debayan/desktop/InLab3/Data/Avengers_Universe`
CORRECT: Relative path: `./Data/Avengers_Universe`

Sample file from `Avengers_Universe`

```
Line1

Line2

Iron_Man

Line 4

...
```

Note :

1. Incase you don’t have Heroes.txt in your Data folder, download it again from moodle.
2. Assume Heroes.txt is inside the Data folder and use its path in your script.

Q4

You all know that IITB follows the credit system, so at the end of every semester we need to calculate the CPI to analyze performance of students.
Write a python program to calculate the CPI of a set of students given their grades in the courses they took and the credits per course.

Create a class named `Student` with,

1. a class variable `num_students` (to keep track of how many `Student` instances have been created) and,
2. instance variables `grades` and `credits` both of type list (of integers), and any extra ones
3. member function `CPI()`, and any extra ones you may require.

Implement a constructor to initialize the variables.
Assume that grades and credits are all positive integers.

Input format:
Grades and credits of each student occur on 2 separate lines.
The string “`---`” acts as a delimiter between students. The final student’s info is NOT followed by the delimiter.

Example:

```
8 6 7 9 8
6 6 6 6 8
---
8 8 8 8 8
6 6 6 6 8
---
10 9 7 7
8 6 6 3
```

Input file is given as argument to the python script.
Usage : python3 Q4.py <input_file>

Output format:
Print the total number of students using the class variable followed by the overall CPI of each student one per line in the order the students were input. Print upto 4 decimal points for the CPI of a student.
(upto 4 decimals implies that you round it to four decimals)

Expected output:

```
3
7.6250
8.0000
8.5652
```

Q5

Read a string as input from `stdin` and find the count of each character in the string and print in each line a character along with its count separated by a colon and a space in decreasing order of frequency, and in case of ambiguity further, by the decreasing ASCII order of the character. The string may have non-alphabetic characters.
Hint: Use dictionary
Input:

```
apple
```

Output:

```
p: 2
l: 1
e: 1
a: 1
```

Since, `l > e > a` as per their ASCII orders.