Python Exercises

Instructions:

1. Go through the exercises and write the solutions.
2. You can go with assumptions if unclear on any specific problem and have your implementations.
3. As and when, you complete exercises, push it to GitHub and share (GitHub id: duraiatpeers)
4. Your code will be reviewed, and the feedback will be shared.

Exercise 1:

Check if a given string is a palindrome or not.

Exercise 2:

Find the first most frequent character in a given string (edited)

Sample output: The given string is : successes
The first most frequent char in the string is: s

Exercise 3:

From the below string extract, IP DATE PICS URL , and print it
Input String
'123.123.123.123 - - [26/Apr/2000:00:23:48 -0400] "GET /pics/wpaper.gif HTTP/1.0" 200 6248 "http://www.jafsoft.com/asctortf/" "Mozilla/4.05 (Macintosh; I; PPC)'

Exercise 4:

Remove the duplicate characters from the String and print it
Sample Output:
The given string is: resources After removing duplicates characters the new string is: resouc

Exercise 5:

Given a String find whether it is a valid 10-digit phone number. Number should be in format xxx-xxx-xxxx E.g 234-456-9999

Exercise 6:

Find whether a number is a Perfect number E.g A perfect number is a positive integer that is equal to the sum of its positive divisors, excluding the number itself. For instance, 6 has divisors 1, 2 and 3 (excluding itself), and 1 + 2 + 3 = 6, so 6 is a perfect number.

Exercise 7

Create a list by picking an odd-index items from the first list and even index items from the second

Given two lists, l1 and l2, write a program to create a third list l3 by picking an odd-index element from the list l1 and even index elements from the list l2.

Given:
l1 = [3, 6, 9, 12, 15, 18, 21]
l2 = [4, 8, 12, 16, 20, 24, 28]

Expected Output:

Element at odd-index positions from list one
[6, 12, 18]

Element at even-index positions from list two
[4, 12, 20, 28]

Printing Final third list
[6, 12, 18, 4, 12, 20, 28]

Exercise 8

Create a Python set such that it shows the element from both lists in a pair. The solution should take two or more iterables (like list, dict, string), aggregates them in a tuple, and returns it.

Given:

first_list = [2, 3, 4, 5, 6, 7, 8]
second_list = [4, 9, 16, 25, 36, 49, 64]

Expected Output:
Result is {(6, 36), (8, 64), (4, 16), (5, 25), (3, 9), (7, 49), (2, 4)} (edited)

Exercise 9

Problem statement:

1. create a data file data.txt with the following data

empid,empname,emplocation,empsalary
e001,iniyal,chennai,20000.00
e002,aniyal,bangalore,25000.00
e003,indulekha,trivandrum,18000.00

2. open the file, read line by line

- extract data and put it in the following format

```
{
"employees": [
    {
      "empid":"e001",
      "empname":"iniyal",
      "emplocation":"chennai"
      "empsalary":20000.00
    },
    {
      "empid":"e002",
      "empname":"iniyal",
      "emplocation":"chennai"
      "empsalary":20000.00

    },{

    }
  ]
}
```

Problem statement:

create a data file data.txt with the following data

empid,empname,emplocation,empsalary
e001,iniyal,chennai,20000.00
e002,aniyal,bangalore,25000.00

1. Get the header columns from the user separated by ,
For e.g empid,empname

empname
empid,empname,emplocation
*
empname,empsalary

Output should be based on the header columns
For e.g, if the userinput is empname, empsalary, then the following data should be printed

2. open the file, read line by line

- extract data and put it in the following format

```
{
"employees": [
    {
      "empname":"iniyal",
```

```
            "empsalary":20000.00
        },
        {
            "empname":"iniyal",
            "empsalary":20000.00

        }
    ]
}
```

Output format:

dict
- list
  - dict

Exercise 10

Use the builtin modules, run dir command and print the filename and the datasize (edited)

Exercise 11

Run the command "netstat -ano" in the command prompt.
Get the output and print the following:

Count of all Unique local address
Count of all Unique Foreign addresses
Unique State
Unique PID

Exercise 12

Given an IP address, check if the IP address is valid or not.
Use regular expressions.

Formats accepted:

0.0.0.0:80
0.0.0.0:135
0.0.0.0:3389
0.0.0.0:49664
127.0.0.1:53
127.0.0.1:6942
127.0.0.1:27017
192.168.1.101:139
192.168.1.101:49601

[::]:80
[::]:135
[::]:7680
[::]:49664
[::1]:53
[::1]:60744
[2401:4900:32f2:4d55:f8a8:d790:9370:86da]:49772

Exercise 13

Generate a 1-D array containing 5 random integers from 0 to 20.
Get input from the user.
if the user input matches the random number, then
    print "Success" and exit
else
    print failure and try again. If the number of chances exceeds 3, then exit.