Rajalakshmi Engineering College

Name: PRASANNA NADRAJAN R
Email: 241801207@rajalakshmi.edu.in
Roll no: 2116241801207
Phone: 8667687296
Branch: REC
Department: I AI & DS FC
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 6_CY

Attempt : 1
Total Mark : 40
Marks Obtained : 40

Section 1 : Coding

1. Problem Statement

Alice is developing a program called "Name Sorter" that helps users organize and sort names alphabetically.

The program takes names as input from the user, saves them in a file, and then displays the names in sorted order.

File Name: sorted_names.txt.

Input Format

The input consists of multiple lines, each containing a name represented as a string.

To end the input and proceed with sorting, the user can enter 'q'.

Output Format

The output displays the names in alphabetical order, each name on a new line.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: Alice Smith
John Doe
Emma Johnson
q
Output: Alice Smith
Emma Johnson
John Doe

Answer

```
# You are using Python
with open("file.txt",'w') as f1:
    data="
    while(True):
        data=input()
        if(data=='q'):
            break
        f1.write(data)
        f1.write("\n")

with open("file.txt","r") as f2:
    data=f2.read()
print("\n".join(sorted(data.split("\n"))))
```

Status : Correct ***Marks : 10/10***

2. Problem Statement

Bob, a data analyst, requires a program to automate the process of analyzing character frequency in a given text. This program should allow the user to input a string, calculate the frequency of each character within the text, save these character frequencies to a file named

"char_frequency.txt," and display the results.

Input Format

The input consists of the string.

Output Format

The first line prints "Character Frequencies:".

The following lines print the character frequency in the format: "X: Y" where X is the character and Y is the count.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: aaabbbccc

Output: Character Frequencies:
a: 3
b: 3
c: 3

Answer

```
# You are using Python
from collections import Counter

s=input().strip()
print("Character Frequencies:")
dd=dict(Counter(s))
for key in dd:
    print(f"{key}: {dd[key]}")
```

Status : Correct ***Marks : 10/10***

3. Problem Statement

Write a program to read the Register Number and Mobile Number of a student. Create user-defined exception and handle the following:

If the Register Number does not contain exactly 9 characters in the specified format(2 numbers followed by 3 characters followed by 4 numbers) or if the Mobile Number does not contain exactly 10 characters, throw an IllegalArgumentException. If the Mobile Number contains any character other than a digit, raise a NumberFormatException.If the Register Number contains any character other than digits and alphabets, throw a NoSuchElementException.If they are valid, print the message ‘valid’ or else print an Invalid message.

Input Format

The first line of the input consists of a string representing the Register number.

The second line of the input consists of a string representing the Mobile number.

Output Format

The output should display any one of the following messages:

If both numbers are valid, print "Valid".

If an exception is raised, print "Invalid with exception message: ", followed by the specific exception message.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 19ABC1001
9949596920

Output: Valid

Answer

```
# You are using Python
class illegalArgumentException(Exception):
    def __init__(self,message):
        self.message=message
        super().__init__(Exception)

class NumberFormatException(Exception):
```

```
    def __init__(self,message):
        self.message=message
        super().__init__(Exception)

class NoSuchElementException(Exception):
    def __init__(self,message):
        self.message=message
        super().__init__(Exception)

reg_num=input()
mob_num=input()
try:
    if(len(reg_num)!=9):
        raise illegalArgumentException("Invalid with exception message: Register
Number should have exactly 9 characters.")
    elif(len(mob_num)!=10):
        raise illegalArgumentException("Invalid with exception message: Mobile
Number should have exactly 10 characters.")

except illegalArgumentException as e:
    print(e.message)
    exit()
try:

    for i in range(len(reg_num)):
        if(0<=i<=1 or 5<=i<=7):
            if not(reg_num[i].isdigit()):
                raise NoSuchElementException("Invalid with exception message:
Register Number should have the format: 2 numbers, 3 characters, and 4
numbers.")
        elif(2<=i<=4):
            if not(reg_num[i].isupper()):
                raise NoSuchElementException("Invalid with exception message:
Register Number should have the format: 2 numbers, 3 characters, and 4
numbers.")


    for i in mob_num:
        if not(i.isdigit()):
            raise NumberFormatException("Invalid with exception message: Mobile
Number should only contain digits.")
```

```
except NoSuchElementException as e:
    print(e.message)
    exit()

except NumberFormatException as ee:
    print(ee.message)
    exit()
print("Valid")
```

Status : Correct **Marks : 10/10**

4. Problem Statement

In the enchanted realm of Academia, you, the Academic Alchemist, are bestowed with a magical quill and a parchment to weave the grades of aspiring students into a tapestry of academic brilliance.

The mission is to craft a Python program that empowers faculty members to enter student grades for any two subjects, stores these magical grades in a mystical file, and then, with a wave of your virtual wand, calculates the GPA to unveil the true essence of academic achievement.

Input Format

The input format is a string representing the student's name, any two subjects, and corresponding grades.

After entering grades, they can type 'done' when prompted for the student's name.

Output Format

The output should display the (average of grades) calculated GPA with a precision of two decimal places.

The magical grades will be saved in a mystical file named "magical_grades.txt".

Refer to the sample output for format specifications.

Sample Test Case

Input: Alice
Math
95
English
88
done

Output: 91.50

Answer

```
# You are using Python
dic={}
dat=None
tot=0
while(dat!='done'):
    dat=input()
    if dat[0].isdigit():
        tot+=int(dat)
print('{:.2f}'.format(tot/2))
```

Status : Correct ***Marks : 10/10***