Rajalakshmi Engineering College

Name: SIVASURIYAN .
Email: 240701518@rajalakshmi.edu.in
Roll no:
Phone: 7010603376
Branch: REC
Department: I CSE FF
Batch: 2028
Degree: B.E - CSE

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

Alex is creating an account and needs to set up a password. The program prompts Alex to enter their name, mobile number, chosen username, and desired password. Password validation criteria include:

Length between 10 and 20 characters.At least one digit.At least one special character from !@#$%^&* set. Display "Valid Password" if criteria are met; otherwise, raise an exception with an appropriate error message.

Input Format

The first line of the input consists of the name as a string.

The second line of the input consists of the mobile number as a string.

The third line of the input consists of the username as a string.

The fourth line of the input consists of the password as a string.

Output Format

If the password is valid (meets all the criteria), it will print "Valid Password"

If the password is weak (fails any one or more criteria), it will print an error message accordingly.

Refer to the sample outputs for the formatting specifications.

Sample Test Case

Input: John
9874563210
john
john1#nhoj
Output: Valid Password

Answer

```
class LengthError(Exception):
    pass
class DigitError(Exception):
    pass
class CharError(Exception):
    pass
n = input()
p = int(input())
u = input()
try:
    pas = input().strip()
    d,s = 0,0
    for i in pas:
        if i.isdigit():
            d+=1
        if not i.isalnum():
            s+=1
    if len(pas) < 10 or len(pas) > 20:
        raise LengthError
```

```
    if not any(ch.isdigit() for ch in pas):
        raise DigitError
    if pas.isalnum():
        raise CharError
    print("Valid Password")
except LengthError:
    print("Should be a minimum of 10 characters and a maximum of 20
characters")
except DigitError:
    print("Should contain at least one digit")
except CharError:
    print("It should contain at least one special character")
```

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

2. 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

```
with open("sorted_names.txt","w") as f:
    while True:
        e = input()
        if e == 'q':
            break
        f.write(e.strip()+'\n')
with open("sorted_names.txt","r") as f:
    l = list(line for line in f)
    l.sort()
    for i in l:
        print(i,end='')
```

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

3. 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

```
from collections import Counter
s = input()
f = {}
f = Counter(s)
with open("char_frequency.txt","w") as file:
    for c,v in f.items():
        file.write(str(c+': '+str(v)+'\n'))
with open("char_frequency.txt","r") as file:
    print("Character Frequencies:")
    for i in file:
        print(i,end='')
```

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

4. Problem Statement

A shopkeeper is recording the daily sales of an item for N days, where the price of the item remains the same for all days. Write a program to calculate the total sales for each day and save them in a file named sales.txt that can store the data for a maximum of 30 days. Then, read the file and display the total earnings for each day.

Note: Total Earnings for each day = Number of Items sold in that day × Price of the item.

Input Format

The first line of input consists of an integer N, representing the number of days.

The second line of input consists of N space-separated integers representing the number of items sold each day.

The third line of input consists of an integer M, representing the price of the item that is common for all N days.

Output Format

If the number of days entered exceeds 30 (N > 30), the output prints "Exceeding limit!" and terminates.

Otherwise, the code reads the contents of the file and displays the total earnings for each day on separate lines.

Contents of the file: The total earnings for N days, with each day's earnings appearing on a separate line.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 4
5 10 5 0
20

Output: 100
200
100
0

Answer

```
class LimitError(Exception):
    pass
try:
    n = int(input())
    if n>30:
        raise LimitError
```

```
except LimitError:
    print("Exceeding limit!")
else:
    l = list(map(int,input().split()))
    p = int(input())
    e = []
    for i in l:
        e.append(str(i*p))
    with open("sales.txt","w") as f:
        f.writelines(e)
    with open("sales.txt","r") as f:
        for line in f:
            print(line)
```

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