Rajalakshmi Engineering College

Name: HARI PRASANNA S
Email: 241501064@rajalakshmi.edu.in
Roll no: 241501064
Phone: 9042038178
Branch: REC
Department: I AI & ML FA
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 6_CY

Attempt : 1
Total Mark : 40
Marks Obtained : 36.5

Section 1 : Coding

1. 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

```
text=input()
frequency={}
for char in text:
    frequency[char]=frequency.get(char,0)+1
with open("char_frequency.txt","w") as file:
    file.write("Character Frequencies:\n")
    for char,count in frequency.items():
        file.write(f"{char}: {count}\n")
print("Character Frequencies:")
for char,count in frequency.items():
    print(f"{char}: {count}")
```

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

2. 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

```
try:
   reg=input().strip()
   mob=input().strip()
   if len(reg)!=9:
      raise ValueError("Register Number should have exactly 9 characters.")
   if not (reg[:2].isdigit() and reg[2:5].isalpha() and reg[5:].isdigit()):
      raise ValueError("Register Number should have the format: 2 numbers,3
characters,a nd 4 numbers.")
   for ch in reg:
      if not(ch.isdigit() or ch.isalpha()):
         raise ValueError("Register Number should only contain digits and
alphabets.")
   if len(mob)!=10:
      raise ValueError("Mobile Number should have exactly 10 characters.")
```

```
    for ch in mob:
        if not ch.isdigit():
            raise ValueError("Mobile Number should only contain digits.")
    print("Valid")
except ValueError as e:
    print(f"Invalid with exception message: {e}")
```

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

3. 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

```
import string

def validate_password(password):
    special_characters = "!@#$%^&*"

    # 1. Check for at least one digit first
    if not any(char.isdigit() for char in password):
        raise Exception("Should contain at least one digit")

    # 2. Check for at least one allowed special character
    if not any(char in special_characters for char in password):
        raise Exception("It should contain at least one special character")

    # 3. Check for length constraint
    if not (10 <= len(password) <= 20):
        raise Exception("Should be a minimum of 10 characters and a maximum of
20 characters")

    # 4. Check for whitespace characters
    if any(char.isspace() for char in password):
        raise Exception("Password should not contain any whitespace characters")

    # 5. Ensure all characters are printable (exclude control or non-ASCII
characters)
    if not all(char in string.printable for char in password):
        raise Exception("Password contains invalid characters")

    print("Valid Password")

# Input
name = input()
mobile = input()
username = input()
password = input()
```

```
# Validation
try:
    validate_password(password)
except Exception as e:
    print(e)
```

Status : Partially correct ***Marks : 6.5/10***

4. 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

```
names=[]
while True:
    name=input()
    if name.lower()=='q':
        break
    names.append(name)
names.sort()
with open("sorted_names.txt","w") as f:
    for name in names:
        f.write(name+ "\n")
with open("sorted_names.txt","r") as fi:
    content=fi.read()
    print(content)
```

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