Rajalakshmi Engineering College

Name: Sarvesh S
Email: 240701479@rajalakshmi.edu.in
Roll no: 240701479
Phone: 9361488694
Branch: REC
Department: I CSE FE
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

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

```
s=input()
fp=open("char_frequency.txt",'w')
fp.write(s)
fp.close()
fp=open("char_frequency.txt",'r')
content=fp.readline()
fp.close()
print("Character Frequencies:")
d={}
for i in content:
   if i not in d:
      d[i]=1
   else:
      d[i]+=1
for i in d:
   print(f"{i}: {d[i]}")
```

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

2. Problem Statement

Implement a program that checks whether a set of three input values can form the sides of a valid triangle. The program defines a function is_valid_triangle that takes three side lengths as arguments and raises a ValueError if any side length is not a positive value. It then checks whether the sum of any two sides is greater than the third side to determine the validity of the triangle.

Input Format

The first line of input consists of an integer A, representing side1.

The second line of input consists of an integer B, representing side2.

The third line of input consists of an integer C, representing side3.

Output Format

The output prints either "It's a valid triangle" if the input side lengths form a valid triangle,

or "It's not a valid triangle" if they do not.

If there is a ValueError, it should print "ValueError: <error_message>".

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 3
4
5
Output: It's a valid triangle

Answer

```
a=int(input())
b=int(input())
c=int(input())
try:
   if a<=0 or b<=0 or c<=0:
      raise ValueError("ValueError: Side lengths must be positive")
   if a+b>c and b+c>a and c+a>b:
      print("It's a valid triangle")
   else:
      print("It's not a valid triangle")
except ValueError as ve:
    print(ve)
```

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

3. 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

```
while(1):
    s=input()
    if s=='q':
        break
    with open("sorted_names.txt",'a') as fp:
        fp.write(s+'\n')
fp=open("sorted_names.txt",'r')
content=fp.readlines()
fp.close()
content.sort()
for i in content:
    print(i)
```

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

4. 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

```
regno=input()
phno=input()
class NumberFormatException(Exception):
   pass
class IllegalArgumentException(Exception):
   pass
class NoSuchElementException(Exception):
   pass
try:
   if not phno.isdigit():
      raise NumberFormatException("Invalid with exception message: Mobile
Number should only contain digits.")
   if len(phno)!=10:
      raise NumberFormatException("Invalid with exception message: Mobile
Number should have exactly 10 characters.")
   if not(regno[:2].isdigit() and regno[2:5].isalpha() and regno[5:].isdigit()):
      raise IllegalArgumentException("Invalid with exception message: Register
Number should have the format: 2 numbers, 3 characters, and 4 numbers.")
   if len(regno)!=9:
      raise IllegalArgumentException("Invalid with exception message: Register
Number should have exactly 9 characters.")
   print("Valid")
except NumberFormatException as e:
   print(e)
except IllegalArgumentException as e:
   print(e)
except NoSuchElementException as e:
   print(e)
```

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