# Rajalakshmi Engineering College

Name: P LAKSHMIKANTHA  REDDY
Email: 240701283@rajalakshmi.edu.in
Roll no: 240701283
Phone: 9600849623
Branch: REC
Department: I CSE FC
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***

```
f=open("char_frequency.txt","a")
a=input()
f.write(a)
f.close()
f=open("char_frequency.txt","r")
s=f.read()
t=''
for i in s:
    if i not in t:
        t+=i
print("Character Frequencies:")
for i in t:
    print(i,":",s.count(i))
f.close()
```

***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***

```
f=open("sorted_names.txt","a")
l=[]
while(True):
    a=input()
    if(a!='q'):
        l.append(a)
    else:
        break
f.writelines(sorted(l))
f.close()
f=open("sorted_names.txt","r")
try:
    for i in range(len(l)):
        print(f.readline())
except EOFError:
```

f.close()

***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***

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

***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***

```
n=int(input())
a=input().split()
c=int(input())
if(n>30):
   print("Exceeding limit!")
else:
   for i in a:
      print(int(i)*c)
```

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