

# Rajalakshmi Engineering College

Name: Monisha S  
Email: 240701335@rajalakshmi.edu.in  
Roll no: 240701335  
Phone: 9444043820  
Branch: REC  
Department: I CSE FD  
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**

```
# You are using Python
n=input()
d={}
for i in n:
    if i in d:
        d[i]+=1
    else:
        d[i]=1
print("Character Frequencies:")
for i in d:
    print(f"{i}: {d[i]}")
```

**Status :** Correct

**Marks :** 10/10

## **2. 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***

# You are using Python

try:

maxi=30

n=int(input())

```
l=list(map(int,input().split()))
p=int(input())
if n>30:
    raise Exception("m")
for i in l:
    print(i*p)
except Exception:
    print("Exceeding limit!")
```

**Status :** Correct

**Marks : 10/10**

### 3. Problem Statement

Write a program to obtain the start time and end time for the stage event show. If the user enters a different format other than specified, an exception occurs and the program is interrupted. To avoid that, handle the exception and prompt the user to enter the right format as specified.

Start time and end time should be in the format 'YYYY-MM-DD HH:MM:SS' If the input is in the above format, print the start time and end time. If the input does not follow the above format, print "Event time is not in the format "

#### ***Input Format***

The first line of input consists of the start time of the event.

The second line of the input consists of the end time of the event.

#### ***Output Format***

If the input is in the given format, print the start time and end time.

If the input does not follow the given format, print "Event time is not in the format".

Refer to the sample output for formatting specifications.

#### ***Sample Test Case***

Input: 2022-01-12 06:10:00  
2022-02-12 10:10:12  
Output: 2022-01-12 06:10:00  
2022-02-12 10:10:12

**Answer**

```
# You are using Python
from datetime import datetime
try:
    s=input()
    s=datetime.strptime(s,"%Y-%m-%d %H:%M:%S")
except:
    print("Event time is not in the format")
    exit()
try:
    e=input()
    e=datetime.strptime(e,"%Y-%m-%d %H:%M:%S")
except:
    print("Event time is not in the format")
    exit()
print(s)
print(e)
```

**Status :** Correct

**Marks :** 10/10

**4. 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***

# You are using Python

try:

```
name=input()
ph=input()
username=input()
password=input()
s=0
d=0
```

```
for i in password:
    if i in "!@#$%^&*":
        s+=1
    elif i in "1234567890":
        d+=1
if(len(password)<9 or len(password)>20):
    raise Exception("l")
if(s==0):
```

```
        raise Exception("s")
    if(d==0):
        raise Exception("d")

    print("Valid Password")

except Exception as e:
    if(str(e)=="d"):
        print("Should contain atleast one digit")
    elif(str(e)=="s"):
        print("It should contain atleast one special character")
    elif(str(e)=="l"):
        print("Should be a minimum of 10 characters and a maximum of 20
characters")
```

**Status :** Correct

**Marks : 10/10**