Rajalakshmi Engineering College

Name: Nikhilesh D
Email: 240701360@rajalakshmi.edu.in
Roll no: 240701360
Phone: 9940617256
Branch: REC
Department: I CSE FD
Batch: 2028
Degree: B.E - CSE

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 6_CY

Attempt : 1
Total Mark : 40
Marks Obtained : 40

Section 1 : Coding

1. 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'.

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

```python
# You are using Python
import sys
lines=sys.stdin.read().splitlines()
if 'q' in lines:
    lines.remove('q')
lines.sort()
with open("Sorted_names.txt","w") as f:
    for name in lines:
        f.write(name+'\n')
with open("Sorted_names.txt","r") as f:
    for line in f:
        print(line.strip())
```

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
n=int(input())
if n>30:
    print("Exceeding limit!")
else:
    items=list(map(int, input().split()))
    price=int(input())
    with open("sales.txt","w") as f:
        for i in range(n):
            f.write(str(items[i]*price)+"\n")

    with open("sales.txt","r") as f:
        for line in f:
            print(line.strip())
```

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

```python
# You are using Python
class IllegalArgumentException(Exception):
    pass
class NumberFormatException(Exception):
    pass
class NoSuchElementException(Exception):
    pass

def validate_reg_no(reg_num):
    if len(reg_num) !=9:
        raise IllegalArgumentException("Register Number should have exactly 9 characters.")
    if not (reg_num[:2].isdigit() and reg_num[2:5].isalpha() and reg_num[5:].isdigit()):
        raise NoSuchElementException("Register Number should have the format: 2 numbers, 3 characters, and 4 numbers.")
def validate_mobile_number(mobile):
    if len(mobile) !=10:
        raise IllegalArgumentException("Mobile Number should have exactly 10 characters.")
    if not mobile.isdigit():
        raise NumberFormatException("Mobile Number should only contain digits.")
try:
    reg_num=input()
    mobile=input()
    validate_reg_no(reg_num)
    validate_mobile_number(mobile)
```

```
    print("Valid")
except Exception as e:
    print("Invalid with exception message:",e)
```

Status : Correct   *Marks : 10/10*

4. 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

```python
# You are using Python
def is_valid_triangle(a,b,c):
    if a<=0 or b<=0 or c<=0:
        raise ValueError("Side lengths must be positive")
    if a+b>c and a+c>b and b+c>a:
        return True
    else:
        return False
try:
    a=int(input())
    b=int(input())
    c=int(input())
    if is_valid_triangle(a,b,c):
        print("It's a valid triangle")
    else:
        print("It's not a valid triangle")
except ValueError as ve:
    print("ValueError:",ve)
```

Status : Correct   *Marks : 10/10*