Rajalakshmi Engineering College

Name: DEVISHREE J
Email: 240701702@rajalakshmi.edu.in
Roll no: 240701702
Phone: 7397766309
Branch: REC
Department: I CSE FF
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 3_CY

Attempt : 1
Total Mark : 30
Marks Obtained : 25

Section 1 : Coding

1. Problem Statement

Raja needs a program that helps him manage his shopping list efficiently.
The program should allow him to perform the following operations:

Add Items: Raja should be able to add multiple items to his shopping list at
once. He will input a space-separated list of items, each item being a string.

Remove Item: Raja should be able to remove a specific item from his
shopping list. He will input the item he wants to remove, and if it exists in
the list, it will be removed. If the item is not found, the program should
notify him.

Update List: Raja might realize he forgot to add some items initially. After
removing unnecessary items, he should be able to update his list by adding
more items. Similar to the initial input, he will provide a space-separated

list of new items.

The first line consists of the initial list of integers should be entered as space-separated values.

The second line consists of the element to be removed should be entered as a single integer value.

The third line consists of the new elements to be appended should be entered as space-separated values.

Output Format

The output displays the current state of Raja's shopping list after each operation. After adding items, removing items, and updating the list, the program prints the updated shopping list in the following format:

"List1: [element1, element2, ... ,element_n]

List after removal: [element1, element2, ... ,element_n]

Final list: [element1, element2, ... ,element_n]".

If the item is not found in the removing item process, print the message "Element not found in the list".

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 1 2 3 4 5
3
6 7 8

Output: List1: [1, 2, 3, 4, 5]
List after removal: [1, 2, 4, 5]
Final list: [1, 2, 4, 5, 6, 7, 8]

Answer

```
a=list(map(int,input().split()))
b=int(input())
c=len(a)
count=0
for i in range(c):
    if(a[i]==b):
        count+=1
        break
print("List1: ",a)
if(count):
    a.remove(b)
    print("List after removal: ",a)
else:
    print("Element not found in the list")
d=list(map(int,input().split()))
e=len(d)
a.extend(d)
print("Final list: ",a)
```

Status : Correct   *Marks : 10/10*

2. Problem Statement

A company is creating email accounts for its new employees. They want to use a naming convention for email addresses that consists of the first letter of the employee's first name, followed by their last name, followed by @company.com.

The company also has a separate email domain for administrative employees.

Write a program that prompts the user for their first name, last name, role, and company and then generates their email address using the appropriate naming convention based on their role. This is demonstrated in the below examples.

Note:

The generated email address should consist of the first letter of the first name, the last name in lowercase, and a suffix based on the role and company, all in lowercase.

Input Format

The first line of input consists of the first name of an employee as a string.

The second line consists of the last name of an employee as a string.

The third line consists of the role of the employee as a string.

The last line consists of the company name as a string.

Output Format

The output consists of a single line containing the generated email address for the employee, following the specified naming convention.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: John
Smith
admin
iamNeo
Output: jsmith@admin.iamneo.com

Answer

```python
# You are using Python
fname=input()
lname=input()
role=input()
company=input()
fini=fname[0].lower()
lname=lname.lower()
role=role.lower()
company=company.lower()
```

```
if role=="admin":
    email=fini+lname+"@admin."+company+".com"
else:
    email=fini+lname+"@"+company+".com"

print(email)
```

Status : Correct   *Marks : 10/10*

3. Problem Statement

Write a program to check if a given string is perfect.

A perfect string must satisfy the following conditions:

The string starts with a consonant.The string alternates between consonants and vowels.Each consonant appears exactly once.Vowels can occur consecutively multiple times but should not be followed immediately by a consonant.

If the string satisfies all these conditions, print "True"; otherwise, print "False".

Input Format

The input consists of a string.

Output Format

The output prints "True" if the string is perfect. Otherwise, print "False".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: capacitor
Output: True

Answer

```python
def is_perfect_string(s):
    vowels=set('aeiou')
    consonants_seen=set()

    if not s or s[0] in vowels:
        return False

    last_char_type='consonant'

    for i,char in enumerate(s):
        if char in vowels:
            if last_char_type=='consonant':
                last_char_type='vowel'

        else:
            if last_char_type=='vowel':

                return False
            if char in consonants_seen:
                return False
            consonants_seen.add(char)
            last_char_type='consonant'

    return True

s=input().strip()
print("True" if is_perfect_string(s) else "False")
```

Status : Partially correct   *Marks : 5/10*