

# Rajalakshmi Engineering College

Name: Sujit S  
Email: 240701545@rajalakshmi.edu.in  
Roll no: 240701545  
Phone: 9444081254  
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 : 30

### Section 1 : Coding

#### 1. Problem Statement

Sarah is a technical writer who is responsible for formatting two important documents. Both documents contain a certain placeholder character that needs to be replaced with another character before they can be finalized. To ensure consistency in formatting, Sarah wants you to help her write a program that processes both documents by replacing the placeholder character with the new one.

Sarah also prefers a neat and structured output, so she wants you to ensure that both modified documents are printed in a single line, separated by a space, using the format() function.

Example

Input:

Hello  
World

o

a

Output:

Hella World

Explanation:

Here the character 'o' is replaced with 'a' in the concatenated string.

### ***Input Format***

The first line contains string1, the first document.

The second line contains string2, the second document.

The third line contains char1, the placeholder character that needs to be replaced.

The fourth line contains char2, the new character that will replace the placeholder.

### ***Output Format***

The output displays a single line containing the modified string1 and string2, separated by a space.

Refer to the sample output for the formatting specifications.

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

Input: Hello  
World

o

a

Output: Hella World

### ***Answer***

```
# You are using Python
a=input()
b=input()
swap=input()
swaped=input()
c=a+b
d=c.replace(swap,swaped)
print(d)
```

**Status :** Correct

**Marks :** 10/10

## 2. 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.

### ***Input Format***

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**

```
lst=list(map(int,input().split()))
remove=int(input())
add=list(map(int,input().split()))
print("List1:",lst)
if remove in lst:
    lst.remove(remove)
    print("List after removal:",lst)
else:
    print("Element not found in the list")
```

```
lst.extend(add)
print("Final list:",lst)
```

**Status :** Correct

**Marks :** 10/10

### 3. 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**

# You are using Python

```
a=input()
```

```
b=input()
```

```
c=input()
```

```
d=input()
```

```
if c.lower()=="admin":
```

```
    print(a[0].lower()+b.lower()+"@"+c.lower()+"."+d.lower()+".com")
```

```
else:
```

```
    print(a[0].lower()+b.lower()+"@"+d.lower()+".com")
```

**Status :** Correct

**Marks : 10/10**