

Rajalakshmi Engineering College

Name: Mithran VT
Email: 240701312@rajalakshmi.edu.in
Roll no:
Phone: 9952919350
Branch: REC
Department: I CSE FC
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

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

```
def generate_email():  
    first_name = input("").strip()  
    last_name = input("").strip()  
    role = input("").strip().lower()  
    company = input("").strip().lower()  
  
    first_initial = first_name[0].lower()  
  
    last_name = last_name.lower()
```

```
if "admin" in role:
    domain = f"admin.{company}.com"
else:
    domain = f"{company}.com"

email = f"{first_initial}{last_name}@{domain}"

print(f"{email}")
```

generate_email()

Status : Correct

Marks : 10/10

2. 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

```
# You are using Python
def is_perfect_string(s):
    s = s.lower()
    vowels = set("aeiou")
    consonants_seen = set()

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

    i = 0
    while i < len(s):
        char = s[i]

        if char not in vowels:
            if char in consonants_seen:
                return False
            consonants_seen.add(char)

            if i + 1 < len(s) and s[i + 1] not in vowels:
                return False
            i += 1

        else:
            while i < len(s) and s[i] in vowels:
                i += 1

            if i < len(s) and s[i] not in vowels:
                return True

    return True

string = input("").strip()
```

```
print(is_perfect_string(string))
```

Status : Correct

Marks : 10/10

3. 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

```
string1 = input()
```

```
string2 = input()
```

```
char1 = input()
```

```
char2 = input()
```

```
modified1 = string1.replace(char1, char2)
```

```
modified2 = string2.replace(char1, char2)
```

```
print("{} {}".format(modified1, modified2))
```

Status : Correct

Marks : 10/10