Rajalakshmi Engineering College

Name: Devadharshini K
Email: 240701626@rajalakshmi.edu.in
Roll no: 2116240701626
Phone: 8248408481
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 : 27.5

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

```
# Read input from the user
first_name = input().strip()
last_name = input().strip()
role = input().strip()
company = input().strip()

# Convert inputs to lowercase for consistency
first_name_lower = first_name.lower()
last_name_lower = last_name.lower()
role_lower = role.lower()
company_lower = company.lower()
```

```
# Determine the email domain based on role
if role_lower == 'admin':
    domain = f"@{company_lower}.admin.com"
else:
    domain = f"@{company_lower}.com"

# Generate the email address
email = f"{first_name_lower[0]}{last_name_lower}{domain}"

# Output the generated email address
print(email)
```

Status : Partially correct ***Marks : 7.5/10***

2. Problem Statement

Raj wants to write a program that takes a list of strings as input and returns the longest word in the list. If there are multiple words with the same length, the program should return the first one encountered.

Help Raj in his task.

Input Format

The input consists of a single line of space-separated strings.

Output Format

The output prints a string representing the longest word in the given list.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: cat dog elephant lion tiger giraffe
Output: elephant

Answer

```
# Read the input line and split into a list of strings
```

```
words = input().split()

# Initialize variables to keep track of the longest word
longest_word = ""
max_length = 0

# Loop through each word to find the longest
for word in words:
    if len(word) > max_length:
        max_length = len(word)
        longest_word = word

# Print the longest word
print(longest_word)
```

Status : Correct ***Marks : 10/10***

3. Problem Statement

You have two strings str1 and str2, both of equal length.

Write a Python program to concatenate the two strings such that the first character of str1 is followed by the first character of str2, the second character of str1 is followed by the second character of str2, and so on.

For example, if str1 is "abc" and str2 is "def", the output should be "adbecf".

Input Format

The input consists of two strings in each line.

Output Format

The output displays the concatenated string in the mentioned format.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: abc

def

Output: adbecf

Answer

```
# Read the two input strings
str1 = input().strip()
str2 = input().strip()

# Initialize an empty string for the result
result = ""

# Loop through the characters of both strings
for i in range(len(str1)):
    result += str1[i] + str2[i]

# Print the concatenated string
print(result)
```

Status : Correct ***Marks : 10/10***