

Subject: Computer Programming: Python

Faculty: Premanand S / SENSE / VIT-CC

FAT Question: 2

Ramesh and Ashwini are best friends in college. They share every detail of their day, and their friend Aditya often teases them as a couple. Being a Python enthusiast, Aditya decides to create a Python program to play the popular FLAMES game to predict the relationship between any two given names.

Task:

Write a Python program that implements the FLAMES game based on the following rules:

Input two names (e.g., "Ramesh" and "Ashwini").

Remove all common characters (case-insensitive) from both names. For example:

For "Ramesh" and "Ashwini", remove common characters like 'a' and 's'.

The remaining characters are "rmeh" and "hwin".

Count the total number of unmatched characters from both names.

Use this count to cyclically eliminate options from the list ["Friends", "Lovers", "Affectionate", "Marriage", "Enemies", "Siblings"] until only one option remains.

Output the result in the format: of .

<girl name> relationship <boy name>

Solution:

Input:

The input for the program is:

Two names: Strings representing the names of two individuals.

Example: "Ramesh" and "Ashwini"

Output:

The output is:

A string indicating the predicted relationship between the two names based on the FLAMES game. Example: "The relationship of Ashwini with Ramesh is: Lovers"

Processing:

The steps to process the input and produce the output include:

1. **Normalization:**
 - Convert both names to lowercase and remove spaces for case-insensitivity and uniformity.
2. **Removing Common Characters:**
 - Compare the two names character by character.
 - Remove matching characters from both names (only one occurrence of each matching character).
3. **Counting Remaining Characters:**
 - Count the total number of unmatched characters from both names after removing common ones.
4. **FLAMES Elimination:**
 - Use the count of unmatched characters to cyclically eliminate options from the FLAMES list:
["Friends", "Lovers", "Affectionate", "Marriage", "Enemies", "Siblings"].
 - Continue until one option remains.

Algorithm:

Algorithm for FLAMES Game:

1. **Input two names.**
2. Normalize names by:
 - Converting both names to lowercase.
 - Removing spaces.
3. Loop through the characters of the first name:
 - For each character in the first name, check if it exists in the second name.
 - If it exists, remove the character from both names.
4. Count the number of remaining characters in both names.
5. Initialize the FLAMES list:

```
flames = ["Friends", "Lovers", "Affectionate", "Marriage", "Enemies", "Siblings"]
```

6. While the length of the FLAMES list is greater than 1:
 - Use the total unmatched character count to determine the elimination index:


```
index = (unmatched_count % len(flames)) - 1
```
 - Remove the element at the index cyclically:
 - If `index >= 0`, split and rejoin the list.
 - If `index < 0`, remove the last element.
7. The remaining element in the FLAMES list is the result.
8. **Output** the final relationship.

Solution Alternative:

While the above method works fine, there are alternative approaches to implementing the FLAMES game:

1. **Using a List for Elimination:**
 - Instead of using modular arithmetic for cyclic elimination, manually iterate through the FLAMES list using a counter.
2. **Pre-compute Results:**
 - Store precomputed FLAMES outcomes for common unmatched character counts in a dictionary to avoid recalculating for each input.
3. **Object-Oriented Design:**
 - Implement the game using a class that encapsulates the operations like normalization, character removal, counting, and FLAMES elimination.
4. **Recursive Elimination:**
 - Use recursion instead of a loop to cyclically eliminate items from the FLAMES list.

Code:

```
def flames_game(name1, name2):
    # Normalize names by converting to lowercase and removing spaces
    name1 = name1.lower().replace(" ", "")
    name2 = name2.lower().replace(" ", "")

    # Remove common characters
    for char in name1[:]:
        if char in name2:
            name1 = name1.replace(char, '', 1)
            name2 = name2.replace(char, '', 1)

    # Count unmatched characters
    total_unmatched = len(name1) + len(name2)

    # FLAMES list
    flames = ["Friends", "Lovers", "Affectionate", "Marriage", "Enemies",
              "Siblings"]

    # Cyclically eliminate options based on the unmatched count
    while len(flames) > 1:
        index = (total_unmatched % len(flames)) - 1
        if index >= 0:
            flames = flames[index + 1:] + flames[:index]
        else:
            flames = flames[:len(flames) - 1]

    # Result
    return flames[0]
```

```
# Input two names
name1 = input("Enter the first name (boy's name): ")
name2 = input("Enter the second name (girl's name): ")

# Play the game
result = flames_game(name1, name2)

# Display result
print(f"The relationship of {name2} with {name1} is: {result}")
```

Output:

Enter the first name (boy's name): ramesh
Enter the second name (girl's name): indhira
The relationship of indhira with ramesh is: Enemies