# MINI PROJECT - 2
## FLAMES Game

- Flames is a popular game named after the acronym friends, lovers, affectionate. marriage, enemies and siblings. this game does not accurately predict whether or not an individual is right for you there are two steps in the game:

1) take two names
2)remove the common characters
3)get the count of the characters that are left
4)take flames letters F,L,A,M,E,S start removing letters using the cunt we gt
5)The letter which lasts the process is the relation.

**Algorithm:**

1. **Input the Names**
    - Read the two names as input.
2. **Remove Common Characters**
    - Compare the characters in both names and remove the common characters.
    - Count the remaining characters after removing common characters.
3. **Calculate Remaining Character Count**
    - Compute the total count of the remaining characters from both names.
4. **FLAMES Process**
    - Start with the word "FLAMES".
    - Use the remaining character count to iteratively remove letters from "FLAMES".
    - Continue this process until only one letter remains.
5. **Determine the Relationship**
    - The remaining letter in "FLAMES" determines the relationship.

**Pseudo code-**

```
Procedure FLAMES_Game(Name1, Name2):
   # Step 1: Normalise Names
   Name1 = RemoveSpaces(Name1)
   Name2 = RemoveSpaces(Name2)
```

```
    Name1 = ConvertToUpper(Name1)
    Name2 = ConvertToUpper(Name2)

    # Step 2: Remove Common Characters
    List1 = ConvertToList(Name1)
    List2 = ConvertToList(Name2)

    For each character in List1:
        If character exists in List2:
            Remove character from both List1 and List2

    # Step 3: Count Remaining Characters
    RemainingCount = Length(List1) + Length(List2)

    # Step 4: FLAMES Elimination Process
    FlamesList = ['F', 'L', 'A', 'M', 'E', 'S']

    While Length(FlamesList) > 1:
        IndexToRemove = (RemainingCount - 1) % Length(FlamesList)
        Remove element at IndexToRemove from FlamesList

    # Step 5: Determine Relationship
    FinalLetter = FlamesList[0]

    Relationship = MapFinalLetterToRelationship(FinalLetter)

    Return Relationship

Function RemoveSpaces(Name):
    Return Name with all spaces removed

Function ConvertToUpper(Name):
    Return Name converted to uppercase

Function ConvertToList(Name):
    Return a list of characters from Name

Function MapFinalLetterToRelationship(Letter):
    If Letter is 'F':
        Return "Friends"
```

```
    Else If Letter is 'L':
        Return "Lovers"
    Else If Letter is 'A':
        Return "Affectionate"
    Else If Letter is 'M':
        Return "Marriage"
    Else If Letter is 'E':
        Return "Enemies"
    Else If Letter is 'S':
        Return "Siblings"
    Else:
        Return "Unknown Relationship"
```

**Code-**

```
def flames_game(name1, name2):
      name1 = name1.replace(" ", "").upper()
    name2 = name2.replace(" ", "").upper()

    # Convert names to lists and remove common characters
    name1_list = list(name1)
    name2_list = list(name2)

    for char in name1_list[:]:
        if char in name2_list:
            name1_list.remove(char)
            name2_list.remove(char)

    # Count remaining characters
    remaining_count = len(name1_list) + len(name2_list)

    # FLAMES calculation
    flames = list("FLAMES")

    while len(flames) > 1:
        index = (remaining_count - 1) % len(flames)
        flames.pop(index)

    # Determine the relation
    relations = {
        'F': 'Friends',
```

```
        'L': 'Lovers',
        'A': 'Affectionate',
        'M': 'Marriage',
        'E': 'Enemies',
        'S': 'Siblings'
    }

    return relations[flames[0]]

# Example usage
name1 = input("Enter the first name: ")
name2 = input("Enter the second name: ")

result = flames_game(name1, name2)
print(f"The result of the FLAMES game is: {result}")
```

## Output: