**Python | Program to implement simple FLAMES game**

Python is a multipurpose language and one can do literally anything with it. Python can also be used for game development. Let’s create a simple FLAMES game without using any external game libraries like PyGame.

**FLAMES**is a popular game named after the acronym: Friends, Lovers, Affectionate, Marriage, Enemies, Sibling. This game does not accurately predict whether or not an individual is right for you, but it can be fun to play this with your friends.

There are two steps in this game:

* Take the two names.
* Remove the common characters with their respective common occurrences.
* Get the count of the characters that are left .
* Take FLAMES letters as [“F”, “L”, “A”, “M”, “E”, “S”]
* Start removing letter using the count we got.
* The letter which last the process is the result.

**Example :**

Input : player1 name : AJAY

player 2 name : PRIYA

Output : Relationship status : Friends

**Explanation:**In above given two names A and Y are common letters which are occurring one time(common count) in both names so we are removing these letters from both names. Now count the total letters that are left here it is 5. Now start removing letters one by one from FLAMES using the count we got and the letter which lasts the process is the result.

Counting is done in anti-clockwise circular fashion.

*FLAMES   
counting is start from F, E is at 5th count so we remove E and start counting again but a this time start from S.   
FLAMS   
M is at 5th count so we remove M and counting start from S.   
FLAS   
S is at 5th count so we remove S and counting start from F.   
FLA   
L is at 5th count so we remove L and counting start from A.   
FA   
A is at 5th count so we remove A. now we have only one letter is remaining so this is the final answer.   
F   
So, the relationship is F i.e. Friends .*

**Approach:** Take two names as input then remove the common characters with their respective common occurrences. For removing purpose we create a user-defined remove\_match\_char function with two arguments as list1 and list2 which stores list of characters of two players name respectively and return list of concatenated list(list1 + “\*” flagst2) and flag value which we store in ret\_list variable.After removing all the common characters, count the total no. of remaining characters then create a result list with FLAMES acronym i.e [“Friends”, “Love”, “Affection”, “Marriage”, “Enemy”, “Siblings”]. Now start removing word one by one until list does not contain only one word, using the total count which we got. the word which remains in the last, is the result.

Below is the implementation:

* Python3

|  |
| --- |
| # function for removing common characters  # with their respective occurrences      def remove\_match\_char(list1, list2):        for i in range(len(list1)):          for j in range(len(list2)):                # if common character is found              # then remove that character              # and return list of concatenated              # list with True Flag              if list1[i] == list2[j]:                  c = list1[i]                    # remove character from the list                  list1.remove(c)                  list2.remove(c)                    # concatenation of two list elements with \*                  # \* is act as border mark here                  list3 = list1 + ["\*"] + list2                    # return the concatenated list with True flag                  return [list3, True]        # no common characters is found      # return the concatenated list with False flag      list3 = list1 + ["\*"] + list2      return [list3, False]      # Driver code  if \_\_name\_\_ == "\_\_main\_\_":        # take first name      p1 = input("Player 1 name : ")        # converted all letters into lower case      p1 = p1.lower()        # replace any space with empty string      p1.replace(" ", "")        # make a list of letters or characters      p1\_list = list(p1)        # take 2nd name      p2 = input("Player 2 name : ")      p2 = p2.lower()      p2.replace(" ", "")      p2\_list = list(p2)        # taking a flag as True initially      proceed = True        # keep calling remove\_match\_char function      # until common characters is found or      # keep looping until proceed flag is True      while proceed:            # function calling and store return value          ret\_list = remove\_match\_char(p1\_list, p2\_list)            # take out concatenated list from return list          con\_list = ret\_list[0]            # take out flag value from return list          proceed = ret\_list[1]            # find the index of "\*" / border mark          star\_index = con\_list.index("\*")            # list slicing perform            # all characters before \* store in p1\_list          p1\_list = con\_list[: star\_index]            # all characters after \* store in p2\_list          p2\_list = con\_list[star\_index + 1:]        # count total remaining characters      count = len(p1\_list) + len(p2\_list)        # list of FLAMES acronym      result = ["Friends", "Love", "Affection", "Marriage", "Enemy", "Siblings"]        # keep looping until only one item      # is not remaining in the result list      while len(result) > 1:            # store that index value from          # where we have to perform slicing.          split\_index = (count % len(result) - 1)            # this steps is done for performing          # anticlock-wise circular fashion counting.          if split\_index >= 0:                # list slicing              right = result[split\_index + 1:]              left = result[: split\_index]                # list concatenation              result = right + left            else:              result = result[: len(result) - 1]        # print final result      print("Relationship status :", result[0]) |

**Output:**

Player 1 name : ANKIT

Player 2 name : DEEPIKA

Relationship status : Marriage

**Code Explanation:**

1. The code starts by taking the first name, player 1 name.
2. It converts all of the letters into lower case and replaces any space with an empty string.
3. Next, it makes a list of all of the letters in player 1’s name.
4. The code then takes the second name, player 2 name.
5. It converts all of the letters into lower case and replaces any space with an empty string.
6. Next, it creates a list of characters from both players’ names.
7. The code starts by setting the proceed flag to True .
8. Then it calls the remove\_match\_char function on each list.
9. The remove\_match\_char function looks for common characters between both lists and removes them.
10. If no common characters are found, then proceed is set to False and the loop ends.
11. However, if common characters are found, then ret\_list[0] stores the index of “\*” in it (the border mark), ret\_list[1] stores the flag value (True ), and star\_index stores the index of where that character was found in p1\_list or p2\_list (depending on which list was used).
12. So after removing all of the common characters from each list, p1\_list and p2\_list
13. The code firstly takes in two player names as input.
14. Next, all of the letters in each name are converted to lower case.
15. Finally, any spaces are replaced with an empty string.
16. A list of these strings is then created.
17. Next, the remove\_match\_char function is called once for each list.
18. This function looks for a common character between the two lists and removes it.
19. If no common characters are found, the return value is [list1, False] and the proceed flag is set to True .
20. If a common character is found, the return value is [list3, True] and the proceed flag is set to False .
21. Finally, the concatenated list is returned with either [list3, True].