Practical 1.

* Title - Set operation

* objectives:
To understand the implementation of list understand the implementation of set interaction & difference etc.

* Problem statement:-
In second year comp. engg. class group A student play cricket, group B student play badminton & group C student play cricket, group B students play Badminton and group C students play football.
Write a python program to implement below functions.
a) A list of students who play both cricket & badminton.
b) List of students who play either cricket or badminton. but not both.
c) Number of students who play neither cricket nor badminton.
d) No. of students who play both cricket and football but not both.

- Outcomes:
  - implemental list.
  - Solve realworld problem of set operation.

- Hardware & Software requirement, programing IDE like VS code etc.

- Theory:

List:

List is a container that stores any type of data in the continuous manner. List are enclosed in the square bracket. In

- Set Theory:

Set is a structure which represent an understand collection of non zero member. mort distinct order objects.

- Set operation

1) Union:

Union means collection of two or all sets.

2) Intersection:

For set A & B, their intersection A ∩ B is

that are in set A & B.

e.g. A = {2, 3, 5}
B = {4, 3, 5}
A ∩ B = {3, 5}

- difference:

Form set A & B difference means removing - subtracting elements of B from A.

eg. A = {4, 5, 6} B = {4, 6, 7}
A − B = {5, 6}

* Algorithm

A)

a) consider i, j moves over elements of list A & B counterm in A & B is set of all elements that are in A but not in B & B resp.

b) start with i = 0.

c) a) start with j = 0 b) Compare element present at ith position of A with element present at jth position of B

d) If it elements is matching then copy it to resultant

e) If j < b continue comparison given in 3.

b) Use increment i to go to step 3.) if i=c then go to step 4.

f) print the list to get common element of A & B.

B) a) Consider i, j to move over elements of A & B

b) i=j=0 check if element at i^{th} position in same at that position of 'c' flag = 0.

d) if it matcher check if its present in B then flag = 1

e) If flag element remains a copy element of that are not present in both A, B in resultant list ~~A∩A∪B~~

f) print list A ∩ B.

C) a) Consider i, j to move over element of A, B consider list A∩A∪B to store element.

b) start with i=0

c) j=0, check if element at ith position is same.

d) check if for 'c'

e) if the element in u matcher with element in A+B then flag=1

f) if flag remains a copy element

in A & B present,
d] Consider to move over
element of A, C & B e.g. P, a, f, h
used for of A, C, A∩C & B
2) i=0
3) a) j=0
b) Check if element of ith position of 'c' flag=0
4) If it matches check if it is present in B then flag is 1
5) If flag remains 0 copy element of A in resultant first of AUB or plain.
6) Increment i to check matching element using step 3 to 5.

Conclusion:
Able to implement different set operation using python list