**PART-A**

**1a. “List1” is a list that contains the “N” different SRN of students read using a user defined function with the help of input(). SRN of “M” more students are to be appended or inserted into “List1” at the appropriate place and also return the index of the SRN entered by user.**

**Algorithm:**

1. Create an empty list named as list1

2. Read the value of n or m

3. Append or store or adding the elements to the list until n or m

4. Increment i by 1 until n or m

5. Display the list1

6. Repeat the step 2-5

7. Read the index value and display the value

**Program :**

**list1 = []**

**n = int(input("Enter number of elements : "))**

**for i in range(0, n):**

**element = int(input())**

**list1.append(element)**

**print(list1)**

**m = int(input("Enter number of elements : "))**

**for i in range(0, m):**

**element1 = int(input())**

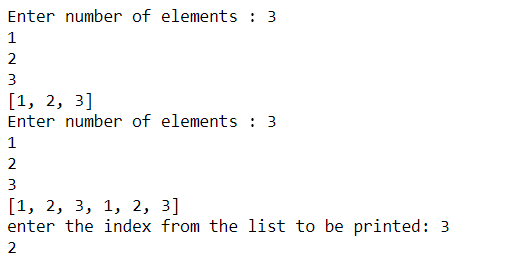
**list1.append(element1)**

**print(list1)**

**p=int(input("enter the index from the list to be printed: "))**

**print(list1.index(p))**

**Output**



**1b. “Tuple1” and “Tuple2” are two tuples that contain “N” different data type read using the user defined function “READ” with the help of input(). Elements of “Tuple1” and “Tuple2”are to be read one at a time and the “larger” value among them should be put into “Tuple3”.**

**Algorithm:**

1. Create an empty tuple named as tuple1
2. Display the empty tuple
3. Add value to the tuple 1
4. Create another tuple2 and add values
5. Display the maximum stored values in tuple1 and tuple2

tuple1 = ()

tuple2 = ()

list1=list(tuple1)

list2=list(tuple2)

count = int(input("Enter the total count of elements :"))

print("enter elements tuple1")

for i in range(0,count):

list1.append(int(input()))

print("enter elemets tuple2")

for i in range(0,count):

list2.append(int(input()))

tuple1=tuple(list1)

tuple2=tuple(list2)

print("tuple1:",tuple1)

print("tuple2:",tuple2)

tuple3 = (max(tuple1),max(tuple2))

print("Maximum or highest number in both tuple1 & tuple2 are",tuple3)

**Output**

Enter the total count of elements :5

enter elements tuple1

10

40

12

89

35

enter elemets tuple2

69

45

17

34

12

tuple1: (10, 40, 12, 89, 35)

tuple2: (69, 45, 17, 34, 12)

Maximum or highest number in both tuple1 & tuple2 are (89, 69)

**2a. Set1 and Set2 are the two sets that contain unique integers. Set3 is to be created by taking the union or intersection of Set1 and Set2using the user defined function Operation(). Perform either union or intersection by reading choice from user. Do not use built in functions union() and intersection() and also the operators “|” and “&“.**

**Algorithm:**

1. Create the set names as people ,vampires and dracula : initialize the values to the created sets
2. Create user defined function union() and perform its operation on any two sets ( i.e people and vampires)
   1. population = people.union (vampires)
3. Display population
4. Apply union operator: repeat step 2.1 and 3
5. Create two set which assigned to set1 and set 2
6. for i in range (5) and add into set1
7. for i in range (3,9) and add into set2
8. Create and Perform user defned function intersection () function on set 1 , set 2 and result stored in set3
9. Dispay set3
10. Apply intersection operation and repeat step 8 & 9

**people = {"Jay", "Idrish", "Archil"}**

**vampires = {"Karan", "Arjun"}**

**dracula = {"Deepanshu", "Raju"}**

# Union using union()

# function

population = people.union(vampires)

print("Union using union() function")

print(population)

# Union using "|"

# operator

population = people|dracula

print("\nUnion using '|' operator")

print(population)

set1 = set()

set2 = set()

for i in range(5):

set1.add(i)

for i in range(3,9):

set2.add(i)

# Intersection using

# intersection() function

set3 = set1.intersection(set2)

print("Intersection using intersection() function")

print(set3)

# Intersection using

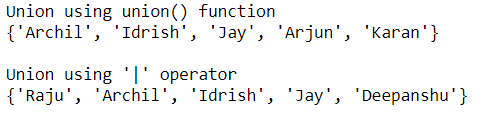
# "&" operator

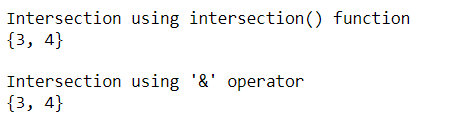
set3 = set1 & set2

print("\nIntersection using '&' operator")

print(set3)

**Output**





**2b. The Dictionary “Dict1” contains N Elements and each element has the operator as the key and operands as values. User reads two operands and an operator and performs the operation on the elements of “Dict1” based on the operator using a user defined function and displays the results.**

**Algorithm:**

1. Create directories with key and values with zip() function
2. Assign directories dict to sampleDict
3. Display SampleDict
4. Create two directories dict1 and dict 2 : Assign key and values
5. Merge dict3 = {\*\*dict1, \*\*dict2}
6. Display dict 3
7. Perform Copy , update , remove and rename the directories key and values
8. Repea step 6 after each function applied on dict3

keys = ['Ten', 'Twenty', 'Thirty']

values = [10, 20, 30]

sampleDict = dict(zip(keys, values))

print(sampleDict)

#Merge following two Python dictionaries into one

dict1 = {'Ten': 10, 'Twenty': 20, 'Thirty': 30}

dict2 = {'Thirty': 30, 'Fourty': 40, 'Fifty': 50}

dict3 = {\*\*dict1, \*\*dict2}

print(dict3)

dict1 = {'Ten': 10, 'Twenty': 20, 'Thirty': 30}

dict2 = {'Thirty': 30, 'Fourty': 40, 'Fifty': 50}

dict3 = dict1.copy()

dict3.update(dict2)

print(dict3)

#Delete set of keys from Python Dictionary

sampleDict = {

"name": "Kelly",

"age":25,

"salary": 8000,

"city": "New york"

}

keysToRemove = ["name", "salary"]

sampleDict = {k: sampleDict[k] for k in sampleDict.keys() - keysToRemove}

print(sampleDict)

#Rename key city to location in the following dictionary

sampleDict = {

"name": "Kelly",

"age":25,

"salary": 8000,

"city": "New york"

}

**Output**

{'Ten': 10, 'Twenty': 20, 'Thirty': 30}

{'Ten': 10, 'Twenty': 20, 'Thirty': 30, 'Fourty': 40, 'Fifty': 50}

{'Ten': 10, 'Twenty': 20, 'Thirty': 30, 'Fourty': 40, 'Fifty': 50}

{'city': 'New york', 'age': 25}