Mode of submission and timings:

1. Question 2 Design (upload the pdf of the shared google document in EduServer) - **3:50 PM**
2. Question 2 Implementation (upload the zipped file of the source code in EduServer) - **4:50 PM**
   - The submission link will be disabled at **5:00 PM**.

Mark distribution

- Question 2: 8 Marks (Design - 4 marks, Implementation and Test cases - 4 marks)

---

2. In set theory, two sets S_i and S_j are said to be **disjoint** if $S_i \cap S_i = \phi$. Let C be a collection of m **disjoint sets** given as $C = \{S_1, S_2, ..S_m\}$, where each S_i is a set of integers.

   Represent each set S_i using a *Binary Search Tree (BST)* and represent the collection C using a *Singly Linked List* L. Each node x in L should be an object with a field *key* and a field *next*, where $x.key$ points to the root node of a BST (representing a set) and $x.next$ points to the next node in the linked list L.

   Write a program that implements the following operations on the collection C as per the function prototypes given below (it may be noted that the sets are **disjoint**):

   You may modify the function prototypes if required.

   - $main()$: Repeatedly read a character 'i', 'f', 'd', 'm', or 'p' from console and call the sub-functions appropriately until character 'e' is encountered.
   - $InsertSet(C, S)$: Insert a new set S into the collection C by inserting a new node (*key* field of the new node should point to the root of the BST representing the set S) to the **front of the linked list** (as the first node) corresponding to the collection C.
   - $FindSet(C, k)$: Find the set S that contains the element k from the collection C. If found, print the element in the root node of the BST corresponding to the set S and return a pointer to this root node. Otherwise, print -1 and return NIL.
   - $DeleteSet(C, k)$: Delete the set S that contains the element k from the collection C. **Note:** Use $FindSet()$ function to locate the set S containing k.
   - $MergeSets(C, k_1, k_2)$:
     1. Find the sets S_i and S_j containing the elements k_1 and k_2, respectively, from the collection C using the function $FindSet()$.
     2. Merge the elements of the set S_i and the set S_j such that the set S_i contains all the elements of both the sets S_i and S_j.
     3. Delete the set S_j that contained k_2 from the collection C.
   - $PrintCollection(C)$: Print all the elements of each set in the collection C by doing an *in-order traversal* of the corresponding tree, starting from the front of the linked list, in sequence. The elements of each set should be printed in a new line and each element should be separated by a space.

Design Instructions

- Write the design for the functions $InsertSet()$, $FindSet()$, $DeleteSet()$, $MergeSets()$, and $PrintCollection()$ only.
- The design for the functions for the *Binary Search Tree* operations like Insert, Delete, Search, and In-order traversal need not be written. You may invoke the functions for the above operations appropriately.

Input/Output Format

The input consists of multiple lines. Each line may contain a character from { 'i', 'd', 'f', 'm', 'p', 'e'} followed by zero or more integers. The integers, if given, are in the range $[1, 10^6]$.

- Character 'i' : Character 'i' will be followed by an integer n. The next line contains n integers separated by a space. Create a new *BST* with the given n integers. Insert the root of the new *BST* into the collection C using the function $InsertSet()$.
- Character 'd' : Character 'd' will be followed by an integer k. Delete the set S_j containing the element k from the collection C using the function $DeleteSet()$.
- Character 'f' : Character 'f' will be followed by an integer k. Find the set S containing the element k from the collection C using the function $FindSet()$.
- Character 'm' : Character 'm' will be followed by two integers k_1 and k_2. Merge the sets S_i and S_j containing the elements k_1 and k_2 respectively using the function $MergeSets()$.
- Character 'p' : Print all the contents of the collection C using the $PrintCollection()$ function.
- Character 'e' : End the program.

Sample Input:

```
i 5
64 32 35 54 12
i 4
87 55 22 98
i 3
45 7 90
i 6
1 2 3 4 5 6
d 55
f 22
f 3
p
d 35
i 4
99 42 15 9
d 2
f 2
p
e
```

Output:

```
87
-1
1
1 2 3 4 5 6
7 45 90
12 32 35 54 64
64
1
-1
9 15 42 99
7 45 90
```