#include<iostream>

using namespace std;

struct Node {

    int data;

    Node\* next;

};

class LinkedList {

public:

    Node\* head;

    LinkedList() {

        head = NULL;

    }

    void insert(int val) {

        Node\* newNode = new Node();

        newNode->data = val;

        newNode->next = head;

        head = newNode;

    }

    bool search(int val) {

        Node\* temp = head;

        while (temp != NULL) {

            if (temp->data == val)

                return true;

            temp = temp->next;

        }

        return false;

    }

    void display() {

        Node\* temp = head;

        while (temp != NULL) {

            cout << temp->data << " ";

            temp = temp->next;

        }

        cout << endl;

    }

    void intersection(LinkedList& set2) {

        Node\* temp = head;

        cout << "Set of students who like both vanilla and butterscotch: ";

        while (temp != NULL) {

            if (set2.search(temp->data)) {

                cout << temp->data << " ";

            }

            temp = temp->next;

        }

        cout << endl;

    }

    void unionSet(LinkedList& set2) {

        Node\* temp = head;

        cout << "Set of students who like either vanilla or butterscotch but not both: ";

        while (temp != NULL) {

            if (!set2.search(temp->data)) {

                cout << temp->data << " ";

            }

            temp = temp->next;

        }

        temp = set2.head;

        while (temp != NULL) {

            if (!search(temp->data)) {

                cout << temp->data << " ";

            }

            temp = temp->next;

        }

        cout << endl;

    }

};

int main() {

    LinkedList vanilla, butterscotch;

    int totalStudents, vanillaCount, butterscotchCount, neitherCount = 0, student;

    cout << "Enter total number of students: ";

    cin >> totalStudents;

    cout << "Enter number of students who like vanilla: ";

    cin >> vanillaCount;

    cout << "Enter the roll numbers of students who like vanilla: ";

    for (int i = 0; i < vanillaCount; i++) {

        cin >> student;

        vanilla.insert(student);

    }

    cout << "Enter number of students who like butterscotch: ";

    cin >> butterscotchCount;

    cout << "Enter the roll numbers of students who like butterscotch: ";

    for (int i = 0; i < butterscotchCount; i++) {

        cin >> student;

        butterscotch.insert(student);

    }

    cout << "\nVanilla set: ";

    vanilla.display();

    cout << "Butterscotch set: ";

    butterscotch.display();

    // a) Set of students who like both vanilla and butterscotch

    vanilla.intersection(butterscotch);

    // b) Set of students who like either vanilla or butterscotch but not both

    vanilla.unionSet(butterscotch);

    // c) Number of students who like neither vanilla nor butterscotch

    cout << "Set of students who like neither vanilla nor butterscotch: ";

    for (int i = 1; i <= totalStudents; i++) {

        if (!vanilla.search(i) && !butterscotch.search(i)) {

            cout << i << " ";

            neitherCount++;

        }

    }

    cout << endl << "Number of students who like neither vanilla nor butterscotch: " << neitherCount << endl;

    return 0;

}

Output

Enter total number of students: 5

Enter number of students who like vanilla: 3

Enter the roll numbers of students who like vanilla: 1 2 3

Enter number of students who like butterscotch: 3

Enter the roll numbers of students who like butterscotch: 3 4 5

Vanilla set: 3 2 1

Butterscotch set: 5 4 3

Set of students who like both vanilla and butterscotch: 3

Set of students who like either vanilla or butterscotch but not both: 2 1 4 5

Set of students who like neither vanilla nor butterscotch:

Number of students who like neither vanilla nor butterscotch: 0