```
/****************************ASSINGMENT 4 **************************
Second year Computer Engineering class, set A of students like Vanilla Ice-cream and set
B of students like butterscotch ice-cream. Write a program to store two sets using a linked
list. compute and display-
a) Set of students who like both vanilla and butterscotch
b) Set of students who like either vanilla or butterscotch or not both
c) Number of students who like neither vanilla nor butterscotch
Name - Tanishq Jaywant Pawar
Roll no- 41
batch- S2
**********************************************************************/
#include <stdio.h>
#include <stdlib.h>


// Node for linked list
struct Node {
  int data;
  struct Node* next;
};


// Create a new node
struct Node* createNode(int data) {
  struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
  newNode->data = data;
  newNode->next = NULL;
  return newNode;
}


// Search element in linked list
int search(struct Node* head, int data) {
  for (struct Node* t = head; t != NULL; t = t->next) {
      if (t->data == data) return 1;
  }
  return 0;
}


// Insert into a SET (avoid duplicates)
void insertSet(struct Node** head, int data) {
  if (!search(*head, data)) {
      struct Node* newNode = createNode(data);
```

```
    newNode->next = *head;
    *head = newNode;
  }
}


// Display linked list
void display(struct Node* head) {
  for (struct Node* t = head; t != NULL; t = t->next)
    printf("%d ", t->data);
  printf("\n");
}


// Intersection: elements present in both
struct Node* intersection(struct Node* A, struct Node* B) {
  struct Node* result = NULL;
  for (struct Node* t = A; t != NULL; t = t->next)
    if (search(B, t->data))
      insertSet(&result, t->data);
  return result;
}


// Symmetric difference: (A\B) ∪ (B\A)
struct Node* symmetricDifference(struct Node* A, struct Node* B) {
  struct Node* result = NULL;
  for (struct Node* t = A; t != NULL; t = t->next)
    if (!search(B, t->data))
      insertSet(&result, t->data);
  for (struct Node* t = B; t != NULL; t = t->next)
    if (!search(A, t->data))
      insertSet(&result, t->data);
  return result;
}


// Union: elements present in A or B
struct Node* setUnion(struct Node* A, struct Node* B) {
  struct Node* result = NULL;
  for (struct Node* t = A; t != NULL; t = t->next)
    insertSet(&result, t->data);
  for (struct Node* t = B; t != NULL; t = t->next)
    insertSet(&result, t->data);
  return result;
}


// Count nodes
int countNodes(struct Node* head) {
  int c = 0;
  for (struct Node* t = head; t != NULL; t = t->next) c++;
  return c;
}


int main() {
  struct Node* vanilla = NULL;
  struct Node* butterscotch = NULL;


  int totalStudents, n1, n2, roll;


  printf("Enter total number of students in class: ");
  scanf("%d", &totalStudents);


  printf("Enter number of students who like Vanilla: ");
  do {
      scanf("%d", &n1);
      if (n1 > totalStudents || n1 < 0)
      printf("Invalid! Enter a number between 0 and %d: ", totalStudents);
     } while (n1 > totalStudents || n1 < 0);


  for (int i = 0; i < n1; i++) {
      scanf("%d", &roll);
      insertSet(&vanilla, roll);
  }


  printf("Enter number of students who like Butterscotch: ");
  do {
      scanf("%d", &n2);
      if (n2 > totalStudents || n2 < 0)
      printf("Invalid! Enter a number between 0 and %d: ", totalStudents);
      } while (n2 > totalStudents || n2 < 0);
  for (int i = 0; i < n2; i++) {
      scanf("%d", &roll);
      insertSet(&butterscotch, roll);
  }


  // a) Students who like both
  printf("\nStudents who like both Vanilla and Butterscotch:\n");
  struct Node* both = intersection(vanilla, butterscotch);
  display(both);


  // b) Students who like either but not both
  printf("Students who like either Vanilla or Butterscotch but not both:\n");
  struct Node* eitherNotBoth = symmetricDifference(vanilla, butterscotch);
  display(eitherNotBoth);


  // c) Students who like neither
  struct Node* uni = setUnion(vanilla, butterscotch);
  int unionCount = countNodes(uni);
  int countNeither = totalStudents - unionCount;
  if (countNeither < 0) countNeither = 0;
  printf("Number of students who like neither Vanilla nor Butterscotch: %d\n",
countNeither);
  return 0;
  }
```