```
//SET OPERATIONS USING LINKED LIST

#include <stdio.h>
#include <stdlib.h>

struct Node
{
   int data;
   struct Node* next;
};

void insert(struct Node** head, int data);
void display(struct Node* head);
void unionSets(struct Node* head1, struct Node* head2, struct Node** unionList);
void intersectionSets(struct Node* head1, struct Node* head2, struct Node** intersectList);
void differenceSets(struct Node* head1, struct Node* head2, struct Node** differenceList);
int isPresent(struct Node* head, int data);

int main()
{
   struct Node* set1 = NULL;
   struct Node* set2 = NULL;
   struct Node* result = NULL;
   int choice, data;

   while (1)
   {
      printf("\nMenu:\n");
      printf("1. Insert in Set 1\n");
      printf("2. Insert in Set 2\n");
      printf("3. Union of Sets\n");
      printf("4. Intersection of Sets\n");
      printf("5. Difference of Sets (Set1 - Set2)\n");
      printf("6. Display Set 1\n");
      printf("7. Display Set 2\n");
      printf("8. Exit\n");
      printf("Enter your choice: ");
      scanf("%d", &choice);

      switch (choice)
      {
         case 1:
            printf("Enter data to insert in Set 1: ");
            scanf("%d", &data);
            insert(&set1, data);
            break;

         case 2:
```

```
            printf("Enter data to insert in Set 2: ");
            scanf("%d", &data);
            insert(&set2, data);
            break;

        case 3:
            result = NULL;
            unionSets(set1, set2, &result);
            printf("Union of Set 1 and Set 2: ");
            display(result);
            break;

        case 4:
            result = NULL;
            intersectionSets(set1, set2, &result);
            printf("Intersection of Set 1 and Set 2: ");
            display(result);
            break;

        case 5:
            result = NULL;
            differenceSets(set1, set2, &result);
            printf("Difference of Set 1 - Set 2: ");
            display(result);
            break;

        case 6:
            printf("Set 1: ");
            display(set1);
            break;

        case 7:
            printf("Set 2: ");
            display(set2);
            break;

        case 8:
            exit(0);

        default:
            printf("Invalid choice! Please try again.\n");
      }
   }
   return 0;
}

void insert(struct Node** head, int data)
{
```

```
    if (isPresent(*head, data))
    {
        printf("Element already present in the set.\n");
        return;
    }
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->next = *head;
    *head = newNode;
    printf("Element inserted.\n");
}

void display(struct Node* head)
{
    struct Node* temp = head;
    if (head == NULL)
    {
        printf("Set is empty.\n");
        return;
    }
    while (temp != NULL)
    {
        printf("%d -> ", temp->data);
        temp = temp->next;
    }
    printf("NULL\n");
}

int isPresent(struct Node* head, int data)
{
    struct Node* temp = head;
    while (temp != NULL)
    {
        if (temp->data == data)
        {
            return 1;
        }
        temp = temp->next;
    }
    return 0;
}

void unionSets(struct Node* head1, struct Node* head2, struct Node** unionList)
{
    struct Node* temp = head1;
    while (temp != NULL)
    {
        insert(unionList, temp->data);
```

```
        temp = temp->next;
    }
    temp = head2;
    while (temp != NULL)
    {
        if (!isPresent(*unionList, temp->data))
        {
            insert(unionList, temp->data);
        }
        temp = temp->next;
    }
}

void intersectionSets(struct Node* head1, struct Node* head2, struct Node** intersectList)
{
    struct Node* temp = head1;
    while (temp != NULL)
    {
        if (isPresent(head2, temp->data))
        {
            insert(intersectList, temp->data);
        }
        temp = temp->next;
    }
}

void differenceSets(struct Node* head1, struct Node* head2, struct Node** differenceList)
{
    struct Node* temp = head1;
    while (temp != NULL)
    {
        if (!isPresent(head2, temp->data))
        {
            insert(differenceList, temp->data);
        }
        temp = temp->next;
    }
}
```

main.c

Share | Run

```c
#include <stdio.h>
#include <stdlib.h>

struct Node
{
    int data;
    struct Node* next;
};

void insert(struct Node** head, int data);
void display(struct Node* head);
void unionSets(struct Node* head1, struct Node* head2, struct
    Node** unionList);
void intersectionSets(struct Node* head1, struct Node* head2,
    struct Node** intersectList);
void differenceSets(struct Node* head1, struct Node* head2,
    struct Node** differenceList);
int isPresent(struct Node* head, int data);
```

Output | Clear

```
8. Exit
Enter your choice: 7
Set 2: 3 -> NULL

Menu:
1. Insert in Set 1
2. Insert in Set 2
3. Union of Sets
4. Intersection of Sets
5. Difference of Sets (Set1 - Set2)
6. Display Set 1
7. Display Set 2
8. Exit
Enter your choice: 3
Element inserted.
Element inserted.
Union of Set 1 and Set 2: 3 -> 7 -> NULL
```