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

struct node {
   struct node *rep;
   struct node *next;
   int data;
   int rank; // Adding rank field
};
struct node *heads[50], *tails[50];
int countRoot = 0;
void makeSet(int x) {
   struct node *new = (struct node *)malloc(sizeof(struct node));
   new->rep = new;
   new->next = NULL;
   new->data = x;
   new->rank = 0;
```

```
    heads[countRoot] = new;
    tails[countRoot++] = new;
}
struct node *find(int a) {
    int i;
    for (i = 0; i < countRoot; i++) {
        struct node *tmp = heads[i];
        while (tmp != NULL) {
            if (tmp->data == a) {
                return tmp->rep;
            }
            tmp = tmp->next;
        }
    }
    return NULL;
}
void unionSets(int a, int b) {
    struct node *rep1 = find(a);
    struct node *rep2 = find(b);
    if (rep1 == NULL || rep2 == NULL) {
        printf("\nElement not present in the DS\n");
        return;
    }
    if (rep1 != rep2) {
        if (rep1->rank < rep2->rank) {
            rep1->rep = rep2;
        } else if (rep1->rank > rep2->rank) {
            rep2->rep = rep1;
```

```
        } else {
            rep2->rep = rep1;
            rep1->rank++;
        }
        struct node *tmp = rep2;
        while (tmp->next != NULL) {
            tmp = tmp->next;
        }
        tmp->next = rep1;
    }
}
int search(int x) {
    int i;
    for (i = 0; i < countRoot; i++) {
        struct node *tmp = heads[i];
        if (tmp->data == x) {
            return 1;
        }
        while (tmp != NULL) {
            if (tmp->data == x) {
                return 1;
            }
            tmp = tmp->next;
        }
    }
    return 0;
}
```

```
int main() {
   int choice, x, y;
   do {
      printf("\n.......MENU.......\n\n1.Make Set\n2.Display set
representatives\n3.Union\n4.Find Set\n5.Exit\n");
      printf("Enter your choice : ");
      scanf("%d", &choice);
      switch (choice) {
         case 1:
            printf("\nEnter new element : ");
            scanf("%d", &x);
            if (search(x) == 1) {
               printf("\nElement already present in the disjoint set DS\n");
            } else {
               makeSet(x);
            }
            break;
         case 2:
            printf("\n");
            for (int i = 0; i < countRoot; i++) {
               printf("%d ", heads[i]->data);
            }
            printf("\n");
            break;
         case 3:
            printf("\nEnter first element : ");
            scanf("%d", &x);
            printf("\nEnter second element : ");
            scanf("%d", &y);
```

```
            unionSets(x, y);
            break;
        case 4:
            printf("\nEnter the element: ");
            scanf("%d", &x);
            struct node *rep = find(x);
            if (rep == NULL) {
                printf("\nElement not present in the DS\n");
            } else {
                printf("\nThe representative of %d is %d\n", x, rep->data);
            }
            break;
        case 5:
            exit(0);
        default:
            printf("\nWrong choice\n");
            break;
      }
   } while (1);
   return 0;
}
```