5.Write a C program to construct a singly linked list by removing duplicate elements in the sorted linked list  
Description:  
Take a sorted list and traverse the list. Compare the current node element with next adjacent node. If it is same then delete second element, if not retain. Finally print the resulting list.  
Sample output:  
Given list {1,2,2,3,3,3,4}  
Resulting list{1,2,3,4}

#include <stdio.h>

struct node{

int data;

struct node \*next;

};

struct node \*head, \*tail = NULL;

void addNode(int data) {

struct node \*newNode = (struct node\*)malloc(sizeof(struct node));

newNode->data = data;

newNode->next = NULL;

if(head == NULL) {

head = newNode;

tail = newNode;

}

else {

tail->next = newNode;

tail = newNode;

}

}

void removeDuplicate() {

struct node \*current = head, \*index = NULL, \*temp = NULL;

if(head == NULL) {

return;

}

else {

while(current != NULL){

temp = current;

index = current->next;

while(index != NULL) {

if(current->data == index->data)

{

temp->next = index->next;

}

else {

temp = index;

}

index = index->next;

}

current = current->next;

}

}

}

void display() {

struct node \*current = head;

if(head == NULL) {

printf("List is empty \n");

return;

}

while(current != NULL) {

printf("%d ", current->data);

current = current->next;

}

printf("\n");

}

int main()

{

addNode(1);

addNode(2);

addNode(3);

addNode(2);

addNode(2);

addNode(4);

addNode(1);

printf("Originals list: \n");

display();

removeDuplicate();

printf("List after removing duplicates: \n");

display();

return 0;

}

**Ouptput:**

