**C Program to remove duplicates from a linked list**

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node\* next;

};

/\* Function to insert a node \*/

void insert\_elements(struct node\*\* head, int new\_data)

{

struct node\* new\_node = (struct node\*) malloc(sizeof(struct node));

new\_node -> data = new\_data;

new\_node -> next = (\*head);

(\*head) = new\_node;

}

/\* Function to print nodes \*/

void display\_list(struct node \*node)

{

while (node!=NULL)

{

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

node = node -> next;

}

}

/\* Function to remove duplicates from a sorted list \*/

void remove\_duplicate\_elements(struct node\* head)

{

struct node\* current = head;

struct node\* next\_next;

if (current == NULL)

return;

while (current -> next != NULL)

{

/\* Compare current node with its next \*/

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

{

next\_next = current -> next -> next;

free(current -> next);

current -> next = next\_next;

}

else

{

current = current -> next;

}

}

}

int main()

{

struct node\* head = NULL;

int n;

printf("Enter the total number of elements : ");

scanf("%d", &n);

printf("\nEnter the sorted linked list : ");

int i;

for(i = 0; i< n; i++)

{

int data;

scanf("%d", &data);

insert\_elements(&head, data);

}

printf("\nLinked list before removing duplicates : ");

display\_list(head);

printf("\n");

remove\_duplicate\_elements(head);

printf("\nLinked list after removing duplicates : ");

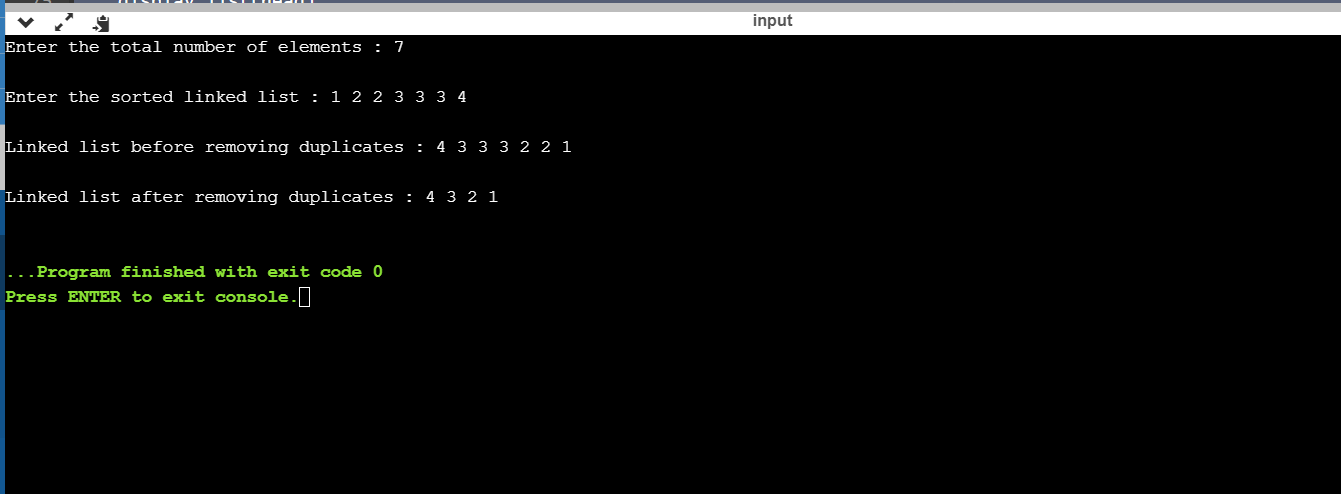
display\_list(head);

printf("\n");

return 0;

}

**OUTPUT**

****