**#5. Linked List**

**Roll Number: cb.en.p2ebs22006**

**Date of Submission: 25-11-2022**

**Aim:**

To perform following operations in a Linked List using C Programming:

1. Creation of linked list
2. Addition of nodes in the list (start, end and middle)
3. Deletion of nodes in the list (start, end and middle)
4. Print the entire list after each of the above operation

**Tools Required:**

Text editor with C Compiler.

**Experiment:**

Code:

#include <stdio.h>

#include<stdlib.h>

typedef struct LinkedList{

int data;

struct LinkedList\* next;

}L;

L \*head=NULL,\*newNode,\*temp;

L\* CreationOfNode(){

int choice=1;

while(choice){

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

printf("Enter the data:");

scanf("%d",&newNode->data);

if(head==NULL){

head=newNode;

temp=newNode;

}

else{

temp->next=newNode;

temp=newNode;

}

printf("DO YOU WANT TO ENTER THE ELEMENTS(1-Yes or 0-No):");

scanf("%d",&choice);

}

}

void DisplayNode(){

temp=head;

printf("The elements in list are:");

while(temp!=0){

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

if(temp->next!=NULL)

printf("->");

temp=temp->next;

}

printf("\n");

}

void insertAtbeginning(){

temp=head;

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

printf("\nEnter the element needs to be enetered at begin:");

scanf("%d",&newNode->data );

newNode->next=temp;

head=newNode;

}

void insertingAtEnd(){

temp=head;

while(temp->next!=0){

temp=temp->next;

}

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

printf("\nEnter the element which needs to be enetered at the end:");

scanf("%d",&newNode->data);

newNode->next=0;

temp->next=newNode;

}

int countOfElements(){

temp=head;

int count=0;

while(temp!=0){

temp=temp->next;

count++;

}

return count;

}

void insertAtMiddle(){

int ch,count,i=0;

count=countOfElements();

printf("\nThe Linked List has %d elements where the new element needs to be entered:",count);

scanf("%d",&ch);

if(count>=ch){

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

temp=head;

while(i<ch-1){

temp=temp->next;

i++;

}

printf("Enter a element which needs to entered at %d the position:",ch);

scanf("%d",&newNode->data);

newNode->next=temp->next;

temp->next=newNode;

}

else{

printf("Enter a valid number");

}

}

void deletingAtbegining(){

temp=head;

head=head->next;

free(temp);

}

void deletingAtEnd(){

L\* prevNode;

temp=head;

while(temp->next!=0){

prevNode=temp;

temp=temp->next;

}

if(temp==head){

head=NULL;

}

else

{ prevNode->next=NULL;

free(temp);

}

}

void deleteAtRandom(){

int ch,i=0,count;

L\* prevNode;

printf("where do you want to remove an element:");

scanf("%d",&ch);

count=countOfElements();

if(count>=ch){

temp=head;

while(i<ch){

prevNode=temp;

temp=temp->next;

i++;

}

prevNode->next=temp->next;

free(temp);

}

else{

printf("Enter a valid number");

}

}

int main() {

int choice;

while(1){

printf("Enter the choice that you want to perform\n1-Node Creation\n2-Inserting Node at the beginning\n3-Inserting Node at the End\n4-Inserting Node in the middle\n5-Deleting a Node at the beginning\n6-Deleting a Node at the end\n7-Deleting Node in the Middle\n8-Exit\nEnter your choice:");

scanf("%d",&choice);

switch(choice){

case 1: CreationOfNode();

DisplayNode();

break;

case 2: insertAtbeginning();

DisplayNode();

break;

case 3:insertingAtEnd();

DisplayNode();

break;

case 4: insertAtMiddle();

DisplayNode();

break;

case 5: deletingAtbegining();

DisplayNode();

break;

case 6: deletingAtEnd();

DisplayNode();

break;

case 7: deleteAtRandom();

DisplayNode();

break;

case 8: exit;

break;

default: printf("Enter a valid number");

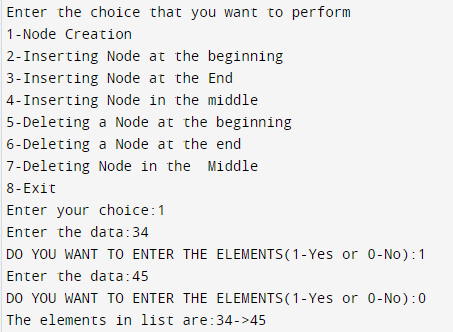
return 0;

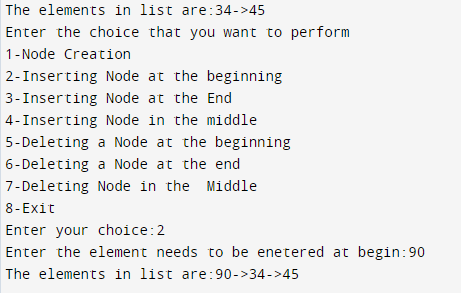
}

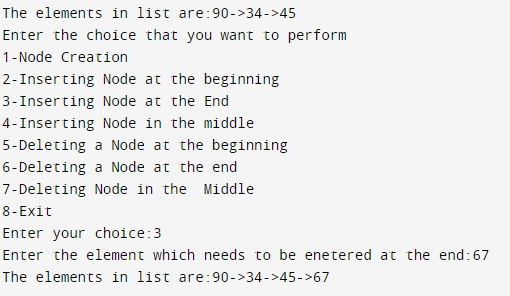
}

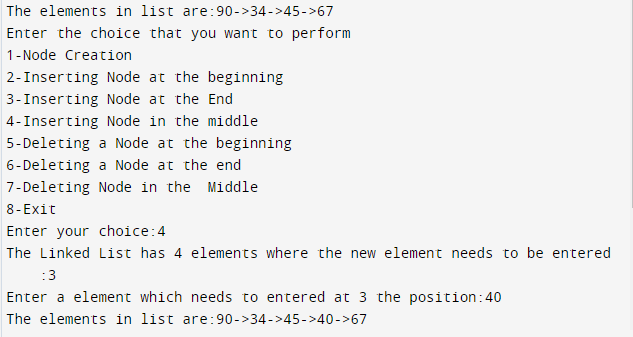
}

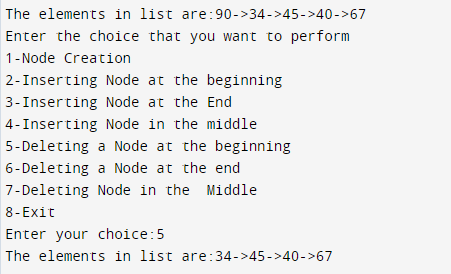
Result:

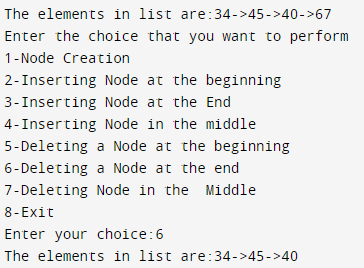


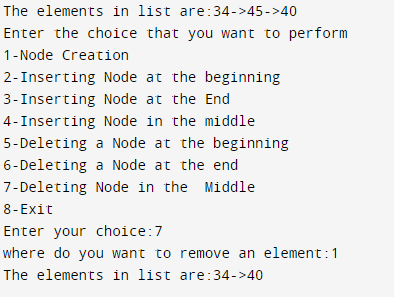












**Inference and Result:**

The Linked List program has been developed and functions such as creating, inserting and dessleting the elements at different positions and the output has been obtained.