**Code:**

#include <iostream>

#include <malloc.h>

using namespace std;

struct node

{

    int data;

    struct node \*next;

};

struct node \*head= NULL;

void insertbegin()

{

    int value;

    struct node \*newnode=NULL;

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

    cout<<"Enter element to be inserted: ";

    cin>>value;

    newnode->data=value;

    if(head==NULL)

    {

        head=newnode;

        head ->next=NULL;

    }

    else

    {

        newnode ->next=head;

        head=newnode;

    }

}

void insertend()

{

    struct node \*newnode;

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

    int value;

    cout<<"Enter elemnt to be inserted: ";

    cin>>value;

    newnode->data=value;

    struct node \*temp=NULL;

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

    temp=head;

    if(head==NULL)

    {

        head=newnode;

        head->next=NULL;

        return;

    }

    while((temp->next)!=NULL)

    {

        temp=temp->next;

    }

    temp->next=newnode;

    newnode->next=NULL;

    cout<<"Value inserted is: "<<newnode->data<<endl;

}

void display()

{

    if(head==NULL)

    {

        cout<<"Linked List is empty: "<<endl;

        return;

    }

    struct node \*temp=NULL;

    temp=head;

    cout<<"Elements in Linked List are: ";

    while(temp!=NULL)

    {

        cout<<temp->data<<" ";

        temp=temp->next;

    }

    cout<<endl;

}

int main()

{

    while(1)

    {

        int choice;

        cout<<"1)Insert Begin 2)Insert End 3)Display: ";

        cin>>choice;

        switch (choice)

        {

        case 1:

            insertbegin();

            break;

        case 2:

            insertend();

            break;

        case 3:

            display();

            break;

        default:

            break;

        }

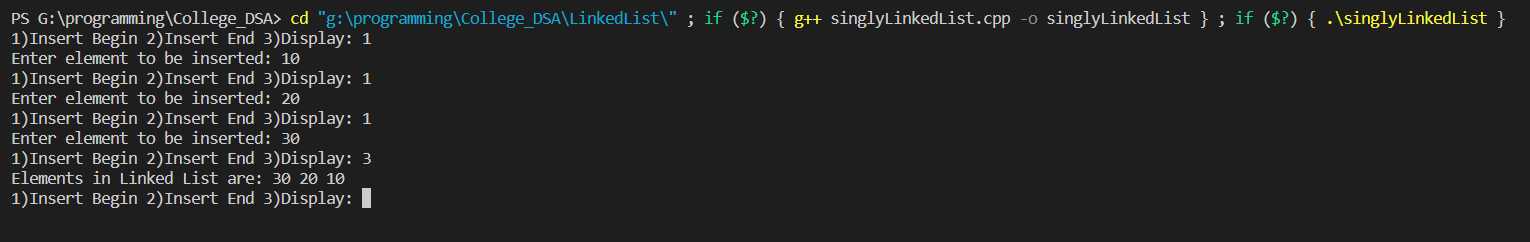
    }

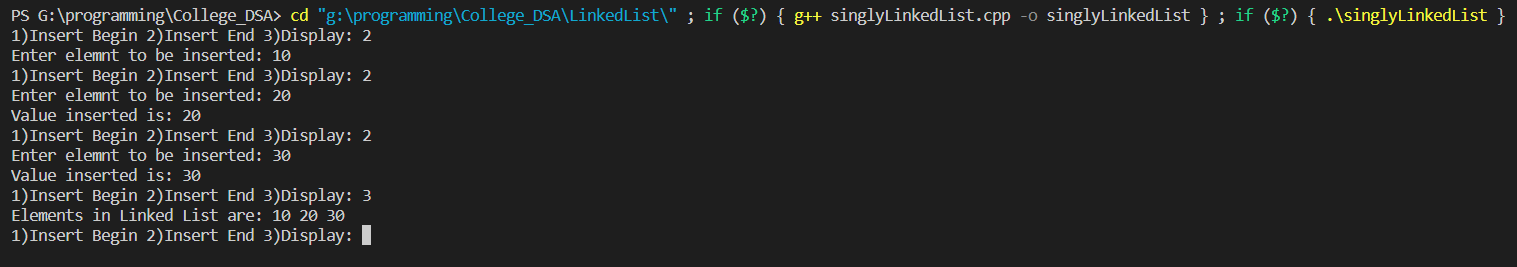
    return 0;

}

**Output:**

**Insert element at begin and display:**

**Inset element at end and display:**

****

**Insert and Display:**

