**AM.SC.P2CSC19034**

**SRUTHY P R**

**COMPITATIVE PROGRAMMING**

Linked List Set 1:

#include<iostream>

#include<stdlib.h>

using namespace std;

class Node

{

public:

int info;

Node\* next;

};

class List:public Node

{

Node \*first,\*last;

public:

List()

{

first=NULL;

last=NULL;

}

void create();

void insert();

void delet();

void display();

void search();

};

void List::create()

{

Node \*temp;

temp=new Node;

int n;

cout<<"\nEnter an Element:";

cin>>n;

temp->info=n;

temp->next=NULL;

if(first==NULL)

{

first=temp;

last=first;

}

else

{

last->next=temp;

last=temp;

}

}

void List::insert()

{

Node \*prev,\*cur;

prev=NULL;

cur=first;

int count=1,pos,ch,n;

Node \*temp=new Node;

cout<<"\nEnter an Element:";

cin>>n;

temp->info=n;

temp->next=NULL;

cout<<"\nINSERT AS\n1:FIRSTNODE\n2:LASTNODE\n3:IN BETWEEN FIRST&LAST NODES";

cout<<"\nEnter Your Choice:";

cin>>ch;

switch(ch)

{

case 1:

temp->next=first;

first=temp;

break;

case 2:

last->next=temp;

last=temp;

break;

case 3:

cout<<"\nEnter the Position to Insert:";

cin>>pos;

while(count!=pos)

{

prev=cur;

cur=cur->next;

count++;

}

if(count==pos)

{

prev->next=temp;

temp->next=cur;

}

else

cout<<"\nNot Able to Insert";

break;

}

}

void List::delet()

{

Node \*prev=NULL,\*cur=first;

int count=1,pos,ch;

cout<<"\nDELETE\n1:FIRSTNODE\n2:LASTNODE\n3:IN BETWEEN FIRST&LAST NODES";

cout<<"\nEnter Your Choice:";

cin>>ch;

switch(ch)

{

case 1:

if(first!=NULL)

{

cout<<"\nDeleted Element is "<<first->info;

first=first->next;

}

else

cout<<"\nNot Able to Delete";

break;

case 2:

while(cur!=last)

{

prev=cur;

cur=cur->next;

}

if(cur==last)

{

cout<<"\nDeleted Element is: "<<cur->info;

prev->next=NULL;

last=prev;

}

else

cout<<"\nNot Able to Delete";

break;

case 3:

cout<<"\nEnter the Position of Deletion:";

cin>>pos;

while(count!=pos)

{

prev=cur;

cur=cur->next;

count++;

}

if(count==pos)

{

cout<<"\nDeleted Element is: "<<cur->info;

prev->next=cur->next;

}

else

cout<<"\nNot Able to Delete";

break;

}

}

void List::display()

{

Node \*temp=first;

if(temp==NULL)

{

cout<<"\nList is Empty";

}

while(temp!=NULL)

{

cout<<temp->info;

cout<<"-->";

temp=temp->next;

}

cout<<"NULL";

}

void List::search()

{

int value,pos=0;

bool flag=false;

if(first==NULL)

{

cout<<"List is Empty";

return;

}

cout<<"Enter the Value to be Searched:";

cin>>value;

Node \*temp;

temp=first;

while(temp!=NULL)

{

pos++;

if(temp->info==value)

{

flag=true;

cout<<"Element"<<value<<"is Found at "<<pos<<" Position";

return;

}

temp=temp->next;

}

if(!flag)

{

cout<<"Element "<<value<<" not Found in the List";

}

}

int main()

{

List l;

int ch;

while(1)

{

cout<<"\n\*\*\*\* MENU \*\*\*\*";

cout<<"\n1:CREATE\n2:INSERT\n3:DELETE\n4:SEARCH\n5:DISPLAY\n6:EXIT\n";

cout<<"\nEnter Your Choice:";

cin>>ch;

switch(ch)

{

case 1:

l.create();

break;

case 2:

l.insert();

break;

case 3:

l.delet();

break;

case 4:

l.search();

break;

case 5:

l.display();

break;

case 6:

return 0;

}

}

return 0;

}

**OUTPUT:**



