ASSIGNMENT 7

AIM : Create a singly linked list for student data and perform :

i) Insertion ii) Deletion iii) Search iv) Modify

OBJECTIVE :

Objective of singly linked list is to store a reference to an object that is an element of the sequence, as well as a reference to the next node of the list. In this assignment, a student database is created using the singly linked list to insert, delete, search and modify the elements in the database.

THEORY :

Singly linked list is a basic linked list type. Singly linked list is a collection of nodes linked together in a sequential way where each node of singly linked list contains a data field and an address field which contains the reference of the next node.

Each node of a singly linked list follows a common basic structure. In a node we can store more than one data fields but we need at least single address field to store the address of next connected node.

struct node

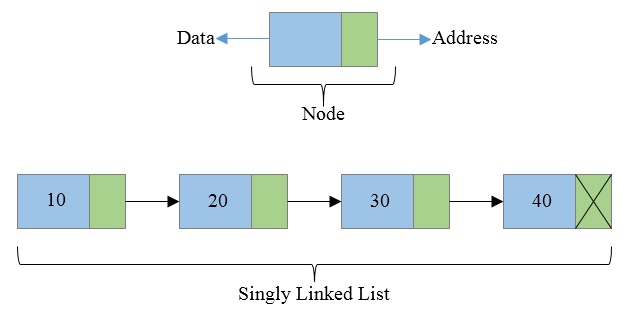
{

int data; // Data

struct node \* next; // Address

};

To perform any operation on a linked list we must keep track/reference of the first node which may be referred by head pointer variable. In singly linked list address field of last node must contain a NULL value specifying end of the list.



Source code :

#include<iostream>

#include<stdlib.h>

using namespace std;

struct node

{

int roll;

char d;

char a[20],b[20],c[20],e[20],f[20];

struct node \*next;

}\*head=NULL;

//Creating the list

void create()

{

struct node \*temp,\*nn;

nn=new node;

cout<<endl;

cout<<"Enter Information of student"<<endl;

cout<<"Enter the Roll: ";

cin>>nn->roll;

cout<<"Enter the surname : ";

cin>>nn->a;

cout<<"Enter the name : ";

cin>>nn->b;

cout<<"Enter Father's name : ";

cin>>nn->c;

cout<<"Enter the div : ";

cin>>nn->d;

cout<<"Enter the contact number : ";

cin>>nn->e;

cout<<"Enter the address of student : ";

cin>>nn->e;

cout<<endl;

nn->next=NULL;

temp=head;

if(head==NULL)

{

head=nn;

}

else

{

while(temp->next!=NULL)

{

temp=temp->next;

}

temp->next=nn;

}

}

//Display the list

void disp()

{

struct node \*temp;

temp=head;

cout<<"Roll"<<" \t\t"<<"Name"<<"\t\t"<<"Division"<<"\t"<<"Address"<<endl;

while(temp!=NULL)

{

cout<<"----------------------------------------------------";

cout<<endl<<temp->roll<<" "<<temp->a<<" "<<temp->b<<" "<<temp->c<<"\t"<<temp->d<<"\t\t"<<temp->e<<endl;

temp=temp->next;

}

}

//Inserting new content in front the list

void insertf()

{

struct node \*nn;

nn=new node;

cout<<endl;

cout<<"Enter Information of student"<<endl;

cout<<"Enter the Roll: ";

cin>>nn->roll;

cout<<"Enter the surname : ";

cin>>nn->a;

cout<<"Enter the name : ";

cin>>nn->b;

cout<<"Enter Father's name : ";

cin>>nn->c;

cout<<"Enter the div : ";

cin>>nn->d;

cout<<"Enter the contact number : ";

cin>>nn->e;

cout<<"Enter the address of student : ";

cin>>nn->e;

cout<<endl;

nn->next=head;

head=nn;

}

//Inserting new content in end the list

void inserte()

{

struct node \*temp,\*nn;

nn=new node;

cout<<endl;

cout<<"Enter Information of student"<<endl;

cout<<"Enter the Roll: ";

cin>>nn->roll;

cout<<"Enter the surname : ";

cin>>nn->a;

cout<<"Enter the name : ";

cin>>nn->b;

cout<<"Enter Father's name : ";

cin>>nn->c;

cout<<"Enter the div : ";

cin>>nn->d;

cout<<"Enter the contact number : ";

cin>>nn->e;

cout<<"Enter the address of student : ";

cin>>nn->e;

cout<<endl;

nn->next=NULL;

temp=head;

while(temp->next!=NULL)

{

temp=temp->next;

}

temp->next=nn;

}

//Inserting new content in any position the list

void insertp()

{

struct node \*temp,\*nn;

int a,i=1;

nn=new node;

cout<<endl;

cout<<"Enter Information of student"<<endl;

cout<<"Enter the Roll: ";

cin>>nn->roll;

cout<<"Enter the surname : ";

cin>>nn->a;

cout<<"Enter the name : ";

cin>>nn->b;

cout<<"Enter Father's name : ";

cin>>nn->c;

cout<<"Enter the div : ";

cin>>nn->d;

cout<<"Enter the contact number : ";

cin>>nn->e;

cout<<"Enter the address of student : ";

cin>>nn->e;

cout<<endl;

nn->next=NULL;

cout<<"Enter position :";

cin>>a;

temp=head;

while(i!=a)

{

temp=temp->next;

i++;

}

nn->next=temp->next;

temp->next=nn;

}

//Deleting content in front the list

void delf()

{

struct node \*temp;

temp=head;

head=temp->next;

cout<<"Deleting First Data"<<endl;

free(temp);

}

//Deleting content in end the list

void dele()

{

struct node \*temp,\*a;

temp=head;

while(temp->next!=NULL)

{

a=temp;

temp=temp->next;

}

a->next=NULL;

cout<<"Deleting Last Data"<<endl;

free(temp);

}

//Deleting content in any position the list

void delp()

{

struct node \*temp,\*a;

int i=1,p;

cout<<"Enter position to delete:";

cin>>p;

temp=head;

while(i!=p)

{

a=temp;

temp=temp->next;

}

a->next=temp->next;

cout<<"Deleting position "<<endl;

free(temp);

}

//Searching content in the list

void search()

{

struct node \*temp;

int k,j=0;

temp=head;

cout<<"\nEnter the roll no. which you want to search : ";

cin>>k;

while(temp!=NULL)

{

if(temp->roll==k)

{

cout<<"Name of roll no. "<<temp->roll<<" is "<<temp->a<<temp->b<<temp->c<<"\n\n";

j++;break;

}

temp=temp->next;

}

if(j==0)

{

cout<<"Roll number is not found "<<endl;

}

}

//modify the content in list

void modify()

{

struct node \*temp;

int k,j=0;

temp=head;

cout<<"\nEnter the roll no. which you want to modify : ";

cin>>k;

while(temp!=NULL)

{

if(temp->roll==k)

{

cout<<"Enter the name of roll no. "<<temp->roll<<" : ";

cin>>temp->a;

j++;break;

}

temp=temp->next;

}

if(j==0)

{

cout<<"Roll number is not found "<<endl;

}

}

int main()

{

int i,n,choice;

cout<<"Enter number of students : ";

cin>>n;

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

{

create();

}

disp();

cout<<"1.Insert at First"<<endl<<"2.Insert at End"<<endl<<"3.Insert at any position"<<"4.Delete at Fisrt"<<endl<<"5.Delete at End"<<endl<<"6.Delete at any position"<<endl<<"7.search"<<endl<<"8.Modify"<<endl;

cin>>choice;

switch(choice)

{

case 1:

insertf();

disp();

break;

case 2:

inserte();

disp();

break;

case 3:

insertp();

disp();

break;

case 4:

delf();

disp();

break;

case 5:

dele();

disp();

break;

case 6:

delp();

disp();

break;

case 7:

search();

break;

case 8:

modify();

break;

default:

cout<<"Invalid option....."<<endl;

break;

}

return 0;

}

Output :

/\*

Enter number of students : 2

Enter Information of student

Enter the Roll: 1

Enter the surname : Gokhe

Enter the name : Nikit

Enter Father's name : Tarachand

Enter the div : D

Enter the contact number : 12345

Enter the address of student : NGP

Enter Information of student

Enter the Roll: 2

Enter the surname : Johanson

Enter the name : Ryan

Enter Father's name : David

Enter the div : A

Enter the contact number : 11111

Enter the address of student : NY

Roll Name Division Address

----------------------------------------------------

1 Gokhe Nikit Tarachand D NGP

----------------------------------------------------

2 Johanson Ryan David A NY

1.Insert at First

2.Insert at End

3.Insert at any position4.Delete at Fisrt

5.Delete at End

6.Delete at any position

7.search

8.Modify

1

Enter Information of student

Enter the Roll: 24

Enter the surname : Joshi

Enter the name : Kartik

Enter Father's name : Suresh

Enter the div : B

Enter the contact number : 22222

Enter the address of student : Pune

Roll Name Division Address

----------------------------------------------------

24 Joshi Kartik Suresh B Pune

----------------------------------------------------

1 Gokhe Nikit Tarachand D NGP

----------------------------------------------------

2 Johanson Ryan David A NY

--------------------------------

Process exited after 121.5 seconds with return value 0

Press any key to continue . . .

\*/