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

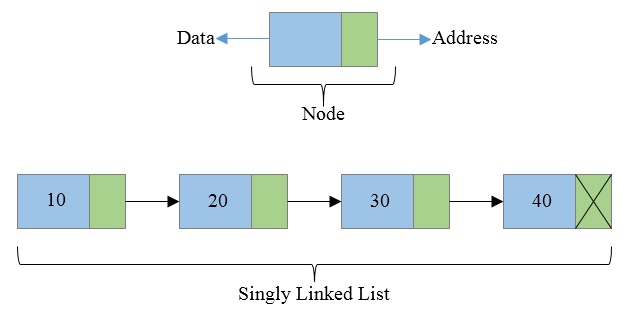
{

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.



ALGORITHM : Algorithm of creation of a Linked List

CREATE---In this algorithm a Linked List of nodes is created. The list is pointed by pointer first, the last node of the list points to NULL., indicating the end of the list.

searching in sLL

1.If first=NULL then{

Print “List empty”; STOP;}

2.ptr=First; [point ptr to the 1st node]

3.while (ptr<>NULL) repeat steps 4 to 5

4.If (DATA (ptr)= ‘X’)

Then {print “item found”;

STOP

}

5.ptr=NEXT (ptr); [shift ptr to the next node]

[end of while]

6.Print “item not found”;

7.END

insertion algo    1.X=new node;

2.Read(DATA(X);

3.If (FIRST=NULL) then

{

First=X;

NEXT(X)=NULL;

}

Else

{

NEXT(X)=First;

First=X;

}

4.END

delete from linked list    1.If (DATA(list)=’VAL’)then

{

Ptr=LIST;

LIST=NEXT(list);

Delete ptr;

Stop;

}

Back=list;

Ptr=list;

2.while(ptr<>NULL) repeat step 3 to 5

3.If(DATA(ptr)=’VAL’) then

{

NEXT(back)=NEXT(ptr);

Delete ptr;

Exit;}

4.back=ptr;

5.ptr=next(ptr);

[end of while loop]

6.END

**Sourcecode:-**

#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;

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;

}

}

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;

}

}

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;

}

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;

}

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;

}

void delf()

{

struct node \*temp;

temp=head;

head=temp->next;

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

free(temp);

}

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);

}

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);

}

void serach()

{

struct node \*temp;

int a;

}

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();

disp1();

break;

default:

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

break;

}

return 0;

}

**Output:-**

**Enter number of students : 1**

**Enter Information of student**

**Enter the Roll: 1**

**Enter the surname : kabadi**

**Enter the name : kaivalya**

**Enter Father's name : sudhir**

**Enter the div : D**

**Enter the contact number : 9874563214**

**Enter the address of student : pune**

**Roll Name Division Address**

**----------------------------------------------------**

**1 kabadi kaivalya sudhir D pune**

**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**

**2**

**Enter Information of student**

**Enter the Roll: 2**

**Enter the surname : Walunjkar**

**Enter the name : Siddhi**

**Enter Father's name : Shahurao**

**Enter the div : D**

**Enter the contact number : 4234**

**Enter the address of student : pune**

**Roll Name Division Address**

**----------------------------------------------------**

**1 kabadi kaivalya sudhir D pune**

**----------------------------------------------------**

**2 Walunjkar Siddhi Shahurao D pune**