**Singly Link List**

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*next;

}\*start=NULL;

void create\_ll();

void display\_ll();

void Insert();

void insert\_first\_ll();

void insert\_last\_ll();

void insert\_nth\_ll();

int search();

void delete();

void count();

void main()

{

printf("----->Enter choice for Link List Operation<-----\n");

int choice=1;

while(choice!=0)

{

printf("\nFor Create press 1\nFor Display press 2\nFor Insert press 3\nFor Search press 4\nFor Delete press 5\nFor Count press 6\nFor Exit press 0\n");

scanf("%d",&choice);

switch(choice)

{

case 1:

printf("\n----->Create<-----\n");

create\_ll();

break;

case 2:

printf("\n----->Display<-----\n");

display\_ll();

break;

case 3:

printf("\n----->Insert<-----\n");

Insert();

break;

case 4:

printf("\n----->Search<-----\n");

search();

break;

case 5:

printf("\n----->Delete<-----\n");

delete();

break;

case 6:

printf("\n----->Count<-----\n");

count();

break;

case 0:

printf("\n----->Exit<-----\n");

break;

default:

printf("Error!Invalid Input\n");

break;

}

}

}

void create\_ll()

{

struct node \*newnode,\*current;

int i,n;

printf("\nEnter numbers of node:\n");

scanf("%d",&n);

for(i=1;i<=n;i++)

{

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

if(newnode==NULL)

{

printf("Error!memory not allocated.");

exit(0);

}

printf("Enter data of node %d\n",i);

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

newnode->next=NULL;

if(start==NULL)

{

start=newnode;

current=newnode;

}

else

{

current->next=newnode;

current=newnode;

}

}

}

void display\_ll()

{

struct node \*temp;

printf("\n\nLink list is:\n");

temp=start;

while(temp!=NULL)

{

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

temp=temp->next;

}

printf("\n");

}

void Insert()

{

printf("\n\nEnter position for insert node:\n");

printf("\nFor first postion press 1\nFor mid/nth position press 2\nFor last position press 3\n");

int ch;

scanf("%d",&ch);

switch(ch)

{

case 1:

printf("----->Insert node at first<-----\n");

insert\_first\_ll();

break;

case 2:

printf("----->Insert node at mid/nth<-----\n");

insert\_nth\_ll();

break;

case 3:

printf("----->Insert node at last<-----\n");

insert\_last\_ll();

break;

default:

printf("Error!Invalid Input\n");

break;

}

}

void insert\_first\_ll()

{

struct node \*newnode;

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

if(newnode==NULL)

{

printf("Error!Memory not allocated successfully.");

exit(0);

}

else

{

printf("\nEnter data of newnode to insert at first:\n");

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

newnode->next=start;

start=newnode;

}

}

void insert\_last\_ll()

{

struct node \*lastnode,\*current;

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

if(lastnode==NULL)

{

printf("Error!Memory not allocated successfully.");

exit(0);

}

else

{

printf("\nEnter data of lastnode to insert at last:\n");

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

lastnode->next=NULL;

current=start;

while(current->next!=NULL)

{

current=current->next;

}

current->next=lastnode;

}

}

void insert\_nth\_ll()

{

struct node \*newnode,\*temp,\*temp1;

int i,pos;

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

if(newnode==NULL)

{

printf("Error! Memory not allocated.");

exit(0);

}

printf("\nEnter data of newnode to insert at nth position:\n");

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

newnode->next=NULL;

printf("Enter position at which node will be inserted:\n");

scanf("%d",&pos);

if(start==NULL)

{

start=newnode;

temp=newnode;

}

else

{

temp=start;

for(i=1;i<pos-1;i++)

{

temp=temp->next;

}

temp1=temp->next;

temp->next=newnode;

newnode->next=temp1;

}

}

int search()

{

struct node \*temp;

int num;

printf("\nEnter data for search:\n");

scanf("%d",&num);

int pos=1,truth=0;

temp=start;

while(temp!=NULL)

{

if(temp->data==num)

{

truth++;

printf("\n%d found at position %d\n",num,pos);

break;

}

temp=temp->next;

pos++;

}

if(truth==0)

printf("%d not found in the list\n",num);

}

void delete()

{

struct node \*temp,\*temp1;

int i,p;

printf("\nEnter position for delete:\n");

scanf("%d",&p);

temp=start;

if(p==1)

{

start=temp->next;

}

else

{

for(i=1;i<p-1;i++)

{

temp=temp->next;

}

}

temp1=temp->next;

temp->next=temp1->next;

}

void count()

{

struct node \*temp;

int length=0;

temp=start;

while(temp!=NULL)

{

length++;

temp=temp->next;

}

printf("\nNumber of node in the link list is %d\n",length);

}