1. Single linked list

#include<stdio.h>

#include<stdlib.h>

struct node{

int data;

struct node\*next;

};

void display();

void insert\_begin();

void insert\_end();

void insert\_pos();

void begin\_delete();

struct node \*head=NULL;

void display()

{

printf("elements are :\n");

struct node \*ptr;

if(head==NULL)

{

printf("list is empty");

return;

}

else{

ptr=head;

while(ptr !=NULL)

{

printf("%d\n", ptr->data);

ptr=ptr->next;

}

}

}

void insert\_begin()

{

struct node\*temp;

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

printf("enter the value to be inserted\n");

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

temp->next=NULL;

if(head==NULL)

head=temp;

else{

temp->next=head;

head=temp;

}

}

void insert\_end()

{

struct node \*temp,\*ptr;

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

printf("enter the value to be inserted \n");

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

temp->next=NULL;

if(head==NULL)

{

head=temp;

}

else

{

ptr=head;

while(ptr->next != NULL)

{

ptr=ptr->next;

}

ptr->next=temp;

}

}

void insert\_pos()

{

int pos,i;

struct node\*temp,\*ptr;

printf("enter the position");

scanf("%d",&pos);

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

printf("enter the value to be inserted\n");

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

temp->next=NULL;

if(pos==0)

{

temp->next=head;

head=temp;

}

else

{

for(i=0, ptr=head; i<pos-1;i++)

{

ptr=ptr->next;

}

temp->next=ptr->next;

ptr->next=temp;

}

}

void begin\_delete()

{

struct node \*ptr;

if(head == NULL)

{

printf("\nList is empty\n");

}

else

{

ptr = head;

head = ptr->next;

free(ptr);

printf("\nNode deleted from the begining ...\n");

}

}

void last\_delete()

{

struct node \*ptr,\*ptr1;

if(head == NULL)

{

printf("\nlist is empty");

}

else if(head -> next == NULL)

{

head = NULL;

free(head);

printf("\nOnly node of the list deleted ...\n");

}

else

{

ptr = head;

while(ptr->next != NULL)

{

ptr1 = ptr;

ptr = ptr ->next;

}

ptr1->next = NULL;

free(ptr);

printf("\nDeleted Node from the last ...\n");

}

}

void random\_delete()

{

struct node \*ptr,\*ptr1;

int loc,i;

printf("\n Enter the location of the node after which you want to perform deletion \n");

scanf("%d",&loc);

ptr=head;

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

{

ptr1 = ptr;

ptr = ptr->next;

if(ptr == NULL)

{

printf("\nCan't delete");

return;

}

}

ptr1 ->next = ptr ->next;

free(ptr);

printf("\nDeleted node %d ",loc+1);

}

void main()

{

int choice;

while(1)

{

printf("\n 1.to insert at the beginning\n"

" 2.to insert at the end\n "

"3.to insert at the position\n "

"4.to display\n "

"5.delete from beginning\n"

"6.delete from end\n"

"7.random delete\n"

"8.exit\n");

printf("enter you choice:\n");

scanf("%d",&choice);

switch(choice)

{

case 1:

insert\_begin();

break;

case 2:

insert\_end();

break;

case 3:

insert\_pos();

break;

case 4:

display();

break;

case 5:

begin\_delete();

break;

case 6:

last\_delete();

break;

case 7:

random\_delete();

break;

case 8:

exit(0);

break;

default:

printf("invalid choice\n");

break;

}

}

}

