Linked List:

1)Implementation of linked list with 3 functions.

#include<stdio.h>

#include<stdlib.h>

void insert();

void delete();

void display();

int ele;

struct node

{

int data;

struct node \*next;

};

struct node \*head;

int main()

{

int ch;

do

{

printf("\n 1.insert");

printf("\n 2.delete");

printf("\n 3.display");

printf("\n 4.exit");

printf("\n enter the choice");

scanf("%d",&ch);

switch(ch)

{

case 1:

insert();

break;

case 2:

delete();

break;

case 3:

display();

break;

case 4:

exit(0);

break;

}

}

while(ch!=4);

return 0;

}

void display()

{

struct node \*tmp;

tmp=head;

if(tmp==NULL)

printf("\n list is empty");

else

{

tmp=head;

while(tmp->next!=NULL)

{

printf("%d ",tmp->data);

tmp=tmp->next;

}

}

}

void delete()

{

struct node \*ptr,\*ptr1;

if(head==NULL)

printf("\n list is empty");

else if(head->next==NULL)

{

head=NULL;

free(head);

printf("\n node is deleted");

}

else

{

ptr=head;

while(ptr->next!=NULL)

{

ptr1=ptr;

ptr=ptr->next;

}

ptr1->next=NULL;

free(ptr);

printf("\n node is deleted");

}

}

void insert()

{

struct node \*ptr,\*tmp;

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

if(ptr==NULL)

printf("\n list is overflow");

else

{

printf("enter the element");

scanf("%d",&ele);

ptr->data=ele;

if(head==NULL)

{

ptr->next=NULL;

head=ptr;

printf("\n node is inserted");

}

else

{

tmp=head;

while(tmp->next!=NULL)

{

tmp=tmp->next;

}

tmp->data=ptr;

ptr->next=NULL;

printf("\n node is inserted");

}

}

}