```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include< stdlib.h>
struct node{
int item; struct node *next;};
typedef struct node *NODE;
NODE getnode(){
NODE x;
x=(NODE)malloc(sizeof(struct node));
return x;}
NODE insert(NODE first, int ~~item~~ data){
NODE n; n=getnode();
n->item=data; n->next=NULL;
if(first==NULL){ return n;}
  n->next=first;
   first=n; return first;}
```

```
NODE del (NODE first){
Node prev, cur;
if(first == NULL){
Printf("Empty \n"); return first;}
cur = first;
while(cur->next != NULL){
prev = cur; cur = cur->next;}
Prev->next = NULL; free(cur);
return first;}

void display (Node first){
int count = 0; Node temp;
if( first == NULL){
printf("Empty \n");}
for(temp = firet; temp!=NULL; temp=temp->next){
Count++; Printf("%d \n", temp->item);}}

Void search( Node first, int data){
int pos = 0; Node temp; int i;
if (first == NULL){
printf("Empty \n"); return;}
for(temp = first, i=0; temp!=NULL; temp=temp->next, i++){
if (temp->item == data){
  Pos = i+1;
  Printf(" Fount @ %d \n", pos); break;}
else{ pos = 0; }}
if (pos == 0){ printf("Not found \n"); }}

void sort (Node first){
int t; Node temp;
if(first == NULL){ printf("Empty \n"); return;}
for(Node i = first; i != NULL; i=i->next){
for(Node j = i->next; j != Null; j=j->next){
if((i->item) > (j->item)){
      t = i->item;
      i->item = j->item;
      j->item = t;}}} printf("Ordered \n");}
```

```
int main () {
Node first = NULL;
Node a = Null; Node b = NULL, Node ans = NULL;
int ch, val, pos, n;
do {
printf("1. Insert \n 2. Del \n 3. Sort \n 4. Search \n 5. Disp \n 6. exit");
scanf("%d", & ch);
~~first = insert (first, val)~~ switch (ch) {
case 1 : printf(" Enter value \n");
scanf("%d, & val); first = insert (first, val); break;
case 2: first = delete (first); break;
case 3: sort (first); break;
case 4: printf(" No. to search \n");
scanf(" %d ", &n); search (first, n); break;
case 5: display (first); break; }} while (choice != 6); }
```