```
#include <stdio.h>
#include<stdlib.h>
typedef struct node{
    int data;
    struct node *link;
}node;
node *root =NULL;
void add_at_bigin ()
{
node *temp;
temp =(node *)malloc(sizeof(node));
printf("Enter Node element \n");
scanf("%d",&temp->data);
temp->link=NULL;
if(root == NULL)
  {
   root =temp
  }
else
 {
  temp->link = root;
   root = temp;
 }
}
```

```
void delete_rear().
{
   struct node *ptr1, *ptr2;
   ptr1 = root;
   while (ptr1 -> link != NULL)
   {
      ptr2 = ptr1;
      ptr1 = ptr1 -> link;
   }
   ptr2 -> link = NULL;
   printf ("Item deleted is %d", ptr1 -> data);
   free (ptr1);
  return;
  }
int length ()
  {
   node *p;
   p = root;
   int i=0;
   whip (p != NULL)
   {
      i++;
      p = p -> link;
   }
  return i;
  }
```

```
void sortlist() {
    struct node * current = root, *inden = NULL;
    int temp;
  if (root == NULL) {
        return;
  }
  else {
    while (current != NULL) {
        inden = current->link;
      while (inden != NULL) {
        if (current->data >= inden->data) {
            temp = current->data;
          current -> data = inden -> data;
           inden -> data = temp;
        }
        inden = inden -> link;
      }
      current = current -> link;
    }
  }
}
```

```
void search()
{
  struct node *temp=root;
  int key;
  int ct=0;
  int flag=0;
  printf("Enter the element need to be searched:");
  scanf("%d",&key);
  while(temp!=NULL)
  {
    ct++;
    printf("element %d is there in position at %d",
           key,ct);
  }
  temp=temp->link;
}
if(flag==0)
  printf("element is not there in list\n");
  return;
}
void display()
{
  node *temp=root;
  if(temp==NULL)
```

```
{
 printf("No nodes in the list \n");
}
else
{
while (temp != NULL)
  {
   printf("%d \n", temp->data);
   temp = temp->link;
  }
}
}

int main()
{
int op, len;
while(1)
printf("\n * * * * * * * \n Enter the operation\n
      1. Insert front \n 2. delete rear \n 3. display\n
       4. Order list \n 5. length\n 6 search\n 7. exit\n");
scanf("%d", &op);
switch(op)
 {
case 1 : add_at_begin();
         break;
```

```
case 2: deleter_rear();
        break;
case 3 : display();
         break;
case 4: sortList();
        break;
case 5 : len = length();
         printf("The length is %d\n", len);
         break;
case 6 : search();
         break;
case 7 : exit(0);
         break;
default :printf("No such operation\n");
}
}
return 0;
}
```