```
#include <stdio.h>
#include <stdlib.h>
struct node
{
 int info;
 struct node * link;
};

typedef struct node* NODE;

NODE getnode()
{
 NODE x;
 x=(NODE) malloc ( sizeof ( struct node));
 if( x==NULL)
  {
   printf ("mem full \n");
   exit (0);
 }
 return x;
}

void freenode (NODE x)
{
 free(x);
}

NODE insert - front (NODE first, int item)
{
  NODE temp;
  temp getnode();
  temp → info  item;
  temp → link NULL;
  if (first   NULL)
    return temp;
  temp → link  first;
  first = temp;
  return first;
}

  NODE delete_rear(NODE first)
  {
   NODE cur, prev;
   if (first == NULL)
   {
    printf("Empty");
   return first;
  }
   if (first → link == NULL)
   {
    printf("item deleted is %d\n", first → info);
   free(first)
   return NULL;
  }
  prev = NULL;
  cur = first;
  while (cur → link != NULL)
   {
```

```
prev = curr;
curr = curr -> link;
}
printf("item deleted at rear-end is %d", curr->info);
free(curr);
~~prev~~ prev -> link = NULL;
   return first;
}

void display (NODE first)
{
 NODE temp;
 if (first == NULL)
 printf("List empty");
 for (temp = first; temp != NULL; temp = temp -> link)
  {
   printf ("%d", temp -> info);
  }
}

void count (NODE first)
{
 NODE temp;
 int count = 1;
 while( temp -> link != NULL)
  {
   temp = temp -> link;
   count + = 1;
}
```

```
    printf("Length of linked list is %d", count);
}

void search(NODE first)
{
    int item, count = 1;
    NODE temp = first;
    printf("enter data to be searched");
    scanf("%d", &item);
    while (temp->info != item)
    {
        temp = temp->link;
        count += 1;
    }
    if (temp != NULL)
    {
        printf("Data is present at %d", count);
    }
    else
    {
        printf("Unsuccessful Search");
    }
}

NODE order list (NODE first)
{
    int swapped, i;
    NODE ptr1, lptr = NULL;
```

```
if(first==NULL)
  return first;

do
  {
  swapped =0;
  ptr1 = first;
  while(ptr1→link != lptr)
    {
      if (ptr1→info > ptr1→link→info)
       {
        int temp1 = ptr1→info;
        ptr1→info = ptr1→link→info;
        ptr1→link→info = temp;
        swapped =1;
       }
     ptr1 = ptr1→link;
   }
   lptr = ptr1;
  }
  while( swapped);
  display(first);
}

int main()
{
  int item, choice,pos,i,n;
  NODE first =NULL;
  for(;;)
{
printf("\n. Insert \n2. Delete - rear \n3. Display \n4. Count.
\n5. Search \n6. Order \n7. Exit");
printf("Enter the choice");
scanf("%d",&choice);
switch(choice);
 {
 case1:
printf("Enter element");
scanf("%d",item);
first = insert_front(first,item);
break;
case2:
first = delete_front_rear(first);
break;
case3:
display(first);
case4:
Count(first);
break;
Case5:
Search(first);
break;
Case6:
first = order_list(first);
break;
Case 7:
exit(0);
default:
  printf("Invalid Input");
  break;
}
}
```