Program to implement LINKED LIST:
insert-front, delete rear, display, count the items ,
Search the items, order the list

main.c

```c
#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;
```

main.c

```
29    temp->info=item;
30    temp->link=NULL;
31    if(first==NULL)
32    return temp;
33    temp->link=first;
34    first=temp;
35    return first;
36    }
37
38    NODE delete_rear(NODE first)
39    {
40    NODE cur,prev;
41    if(first==NULL)
42    {
43    printf("list is empty cannot delete\n");
44    return first;
45    }
46    if(first->link==NULL)
47    {
48    printf("item deleted is %d\n",first->info);
49    free(first);
50    return NULL;
51    }
52    prev=NULL;
53    cur=first;
54    while(cur->link!=NULL)
55    {
56    prev=cur;
57    cur=cur->link;
```

```
57    cur=cur->link;
58    }
59    printf("iten deleted at rear-end is %d",cur->info);
60    free(cur);
61    prev->link=NULL;
62    return first;
63    }
64
65    void display(NODE first)
66    {
67     NODE temp;
68     if(first==NULL)
69     printf("list empty cannot display items\n");
70     for(temp=first;temp!=NULL;temp=temp->link)
71      {
72      printf("%d\n",temp->info);
73      }
74    }
75
76    int length(NODE first)
77    {
78        NODE cur;
79        int count=0;
80        if(first==NULL)
81        return 0;
82        cur=first;
83        while(cur!=NULL)
84        {
85            count++;
```

```
86          cur=cur->link;
87      }
88      return count;
89  }
90
91  void search(int key,NODE first)
92  {
93      NODE cur;
94      if(first==NULL)
95      {
96          printf("List is empty\n");
97          return;
98      }
99      cur=first;
100     while(cur!=NULL)
101     {
102         if(key==cur->info)
103         break;
104         cur=cur->link;
105     }
106     if(cur==NULL)
107     {
108         printf("Search is unsuccessful\n");
109         return;
110     }
111     printf("Search is successful\n");
112 }
113
114 NODE asc(NODE first)
```

main.c

```
114  NODE asc(NODE first)
115  {
116      NODE prev=first;
117      NODE cur=NULL;
118          int temp;
119
120  if(first== NULL) {
121      return 0;
122      }
123  else {
124      while(prev!= NULL) {
125
126      cur = prev->link;
127
128      while(cur!= NULL) {
129
130          if(prev->info > cur->info) {
131      temp = prev->info;
132      prev->info = cur->info;
133      cur->info = temp;
134       }
135       cur = cur->link;
136      }
137      prev= prev->link;
138              }
139          }
140          return first;
141       }
142
```

main.c

```
143         NODE des(NODE first)
144 {
145     NODE prev=first;
146     NODE cur=NULL;
147         int temp;
148
149 if(first==NULL) {
150   return 0;
151   }
152   else {
153     while(prev!= NULL) {
154
155     cur = prev->link;
156
157     while(cur!= NULL) {
158
159       if(prev->info < cur->info) {
160     temp = prev->info;
161     prev->info = cur->info;
162     cur->info = temp;
163         }
164         cur = cur->link;
165         }
166         prev= prev->link;
167          }
168          }
169          return first;
170       }
171
```

```
172  int main()
173  {
174  int item,choice,count,key,option;
175  NODE first=NULL;
176  for(;;)
177  {
178      printf("\n 1:Insert_front\n 2:Delete_rear\n
     3:Display_list\n 4:Count items\n 5:Search items\n
     6:Order_list\n 7:Exit\n");
179      printf("enter the choice\n");
180      scanf("%d",&choice);
181      switch(choice)
182      {
183          case 1:
184              printf("enter the item at front-end\n");
185            scanf("%d",&item);
186            first=insert_front(first,item);
187            break;
188          case 2:
189              first=delete_rear(first);
190            break;
191          case 3:
192              display(first);
193            break;
194          case 4:
195              count=length(first);
196              printf("Length of items in the list is %d\n ",
                 count);
197              break;
```

main.c

```
                count);
197             break;
198         case 5:
199             printf("enter the item to be searched\n");
200             scanf("%d",&key);
201             search(key,first);
202             break;
203         case 6:
204             printf("\n1:ascending ordered_list\n2:descending
                ordered_list\n");
205             scanf("%d",&option);
206             if(option==1)
207             {
208                 first=asc(first);
209                 display(first);
210             }
211             else
212             {
213                 first=des(first);
214                 display(first);
215             }
216             break;
217         default:
218         exit(0);
219     }
220 }
221 return 0;
222 }
```

```
clang-7 -pthread -lm -o main main.c
./main

 1:Insert_front
 2:Delete_rear
 3:Display_list
 4:Count items
 5:Search items
 6:Order_list
 7:Exit
enter the choice
1
enter the item at front-end
23

 1:Insert_front
 2:Delete_rear
 3:Display_list
 4:Count items
 5:Search items
 6:Order_list
 7:Exit
enter the choice
1
enter the item at front-end
34

 1:Insert_front
 2:Delete_rear
 3:Display_list
 4:Count items
 5:Search items
 6:Order_list
 7:Exit
enter the choice
```

```
 4:Count items
 5:Search items
 6:Order_list
 7:Exit
enter the choice
2
item deleted is 34

 1:Insert_front
 2:Delete_rear
 3:Display_list
 4:Count items
 5:Search items
 6:Order_list
 7:Exit
enter the choice
1
enter the item at front-end
2

 1:Insert_front
 2:Delete_rear
 3:Display_list
 4:Count items
 5:Search items
 6:Order_list
 7:Exit
enter the choice
1
enter the item at front-end
3

 1:Insert_front
 2:Delete_rear
 3:Display_list
```

```
enter the choice
1
enter the item at front-end
3

 1:Insert_front
 2:Delete_rear
 3:Display_list
 4:Count items
 5:Search items
 6:Order_list
 7:Exit
enter the choice
3
3
2

 1:Insert_front
 2:Delete_rear
 3:Display_list
 4:Count items
 5:Search items
 6:Order_list
 7:Exit
enter the choice
4
Length of items in the list is 2

 1:Insert_front
 2:Delete_rear
 3:Display_list
 4:Count items
 5:Search items
▸ clang-7 -pthread -lm -o main main.c
▸ ./main
```

```
2
iten deleted at rear-end is 34
 1:Insert_front
 2:Delete_rear
 3:Display_list
 4:Count items
 5:Search items
 6:Order_list
 7:Exit
enter the choice
3
56
45
23

 1:Insert_front
 2:Delete_rear
 3:Display_list
 4:Count items
 5:Search items
 6:Order_list
 7:Exit
enter the choice
4
Length of items in the list is 3

 1:Insert_front
 2:Delete_rear
 3:Display_list
 4:Count items
 5:Search items
 6:Order_list
 7:Exit
enter the choice
```