Logs & others

Start here | ordered_list.c | lab_8_ds.c | lab_7_ds.c | lab_5_ds.c | lab_5_6_ds.c | lab_7.c

```c
/*ordered_list*/
#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_front(NODE first){
NODE temp;
if(first==NULL){
printf("list is empty cannot delete\n");
return first;
}
temp=first;
temp=temp->link;
printf("item deleted at front-end is=%d\n",first->info);
free(first);
return temp;
```

Logs & others

Start here | ordered_list.c | lab_8_ds.c | lab_7_ds.c | lab_5_ds.c | lab_5_6_ds.c | lab_7.c

```
39  temp=first;
40  temp=temp->link;
41  printf("item deleted at front-end is=%d\n",first->info);
42  free(first);
43  return temp;
44  }
45  NODE insert_rear(NODE first,int item){
46  NODE temp,cur;
47  temp=getnode();
48  temp->info=item;
49  temp->link=NULL;
50  if(first==NULL)
51  return temp;
52  cur=first;
53  while(cur->link!=NULL)
54  cur=cur->link;
55  cur->link=temp;
56  return first;
57  }
58  NODE delete_rear(NODE first){
59  NODE cur,prev;
60  if(first==NULL){
61  printf("list is empty cannot delete\n");
62  return first;
63  }
64  if(first->link==NULL){
65  printf("item deleted is %d\n",first->info);
66  free(first);
67  return NULL;
68  }
69  prev=NULL;
70  cur=first;
71  while(cur->link!=NULL){
72  prev=cur;
73  cur=cur->link;
74  }
75  printf("item deleted at rear-end is %d",cur->info);
76  free(cur);
77  prev->link=NULL;
78  return first;
79  }
80  NODE order_list(int item,NODE first){
81  NODE temp,prev,cur;
```

ogs & others

Start here | ordered_list.c | lab_8_ds.c | lab_7_ds.c | lab_5_ds.c | lab_5_6_ds.c | lab_7.c

```
78  return first;
79  }
80  NODE order_list(int item,NODE first){
81  NODE temp,prev,cur;
82  temp=getnode();
83  temp->info=item;
84  temp->link=NULL;
85  if(first==NULL)
86  return temp;
87  if(item<first->info){
88  temp->link=first;
89  return temp;
90  }
91  prev=NULL;
92  cur=first;
93  while(cur!=NULL&&item>cur->info){
94  prev=cur;
95  cur=cur->link;
96  }
97  prev->link=temp;
98  temp->link=cur;
99  return first;
100 }
101 NODE delete_info(int key,NODE first){
102 NODE prev,cur;
103 if(first==NULL){
104 printf("list is empty\n");
105 return NULL;
106 }
107 if(key==first->info){
108 cur=first;
109 first=first->link;
110 freenode(cur);
111 return first;
112 }
113 prev=NULL;
114 cur=first;
115 while(cur!=NULL){
116 if(key==cur->info)
117 break;
118 prev=cur;
119 cur=cur->link;
120 }
```

Logs & others

Start here | ordered_list.c | lab_8_ds.c | lab_7_ds.c | lab_5_ds.c | lab_5_6_ds.c | lab_7.c

```
117     break;
118     prev=cur;
119     cur=cur->link;
120     }
121     if(cur==NULL){
122     printf("search is unsuccessfull\n");
123     return first;
124     }
125     prev->link=cur->link;
126     printf("key deleted is %d",cur->info);
127     freenode(cur);
128     return first;
129     }
130     void display(NODE first){
131     NODE temp;
132     if(first==NULL)
133     printf("list empty cannot display items\n");
134     for(temp=first;temp!=NULL;temp=temp->link){
135     printf("%d\n",temp->info);
136     }
137     }
138     void main(){
139     int item,choice,key;
140     NODE first=NULL;
141     for(;;){
142     printf("\n 1:Insert_front\n 2:Delete_front\n 3:Insert_rear\n 4:Delete_rear\n 5:Order_list\n6:Delete_info\n 7:Display_list\n 8:Exit\n");
143     printf("enter the choice\n");
144     scanf("%d",&choice);
145     switch(choice){
146     case 1:printf("\nenter the item at front-end\n");
147     scanf("%d",&item);
148     first=insert_front(first,item);
149     break;
150     case 2:first=delete_front(first);
151     break;
152     case 3:printf("\nenter the item at rear-end\n");
153     scanf("%d",&item);
154     first=insert_rear(first,item);
155     break;
156     case 4:first=delete_rear(first);
157     break;
158     case 5:printf("\nenter the item to be inserted in ordered_list\n");
159     scanf("%d",&item);
```

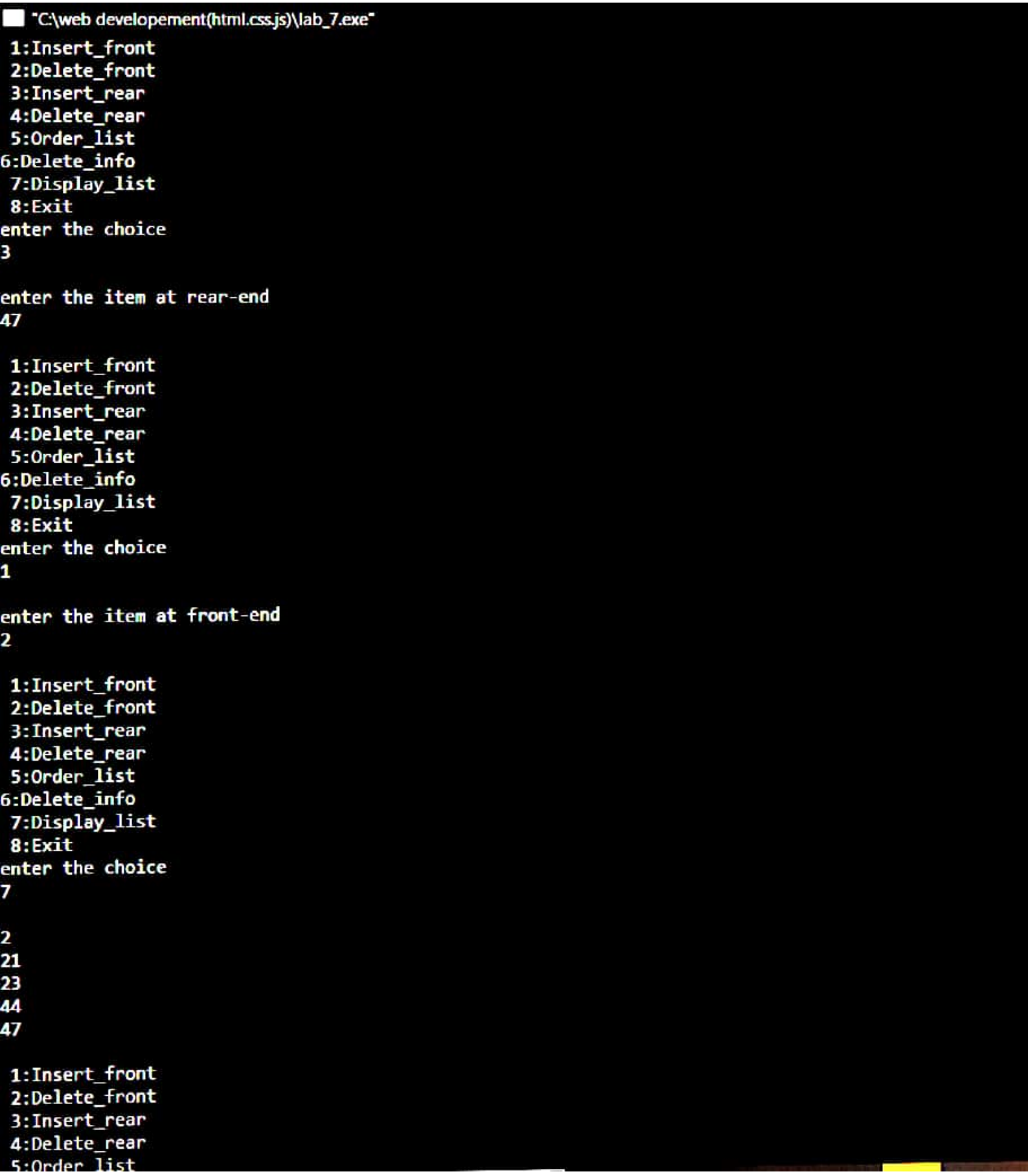
Logs & others

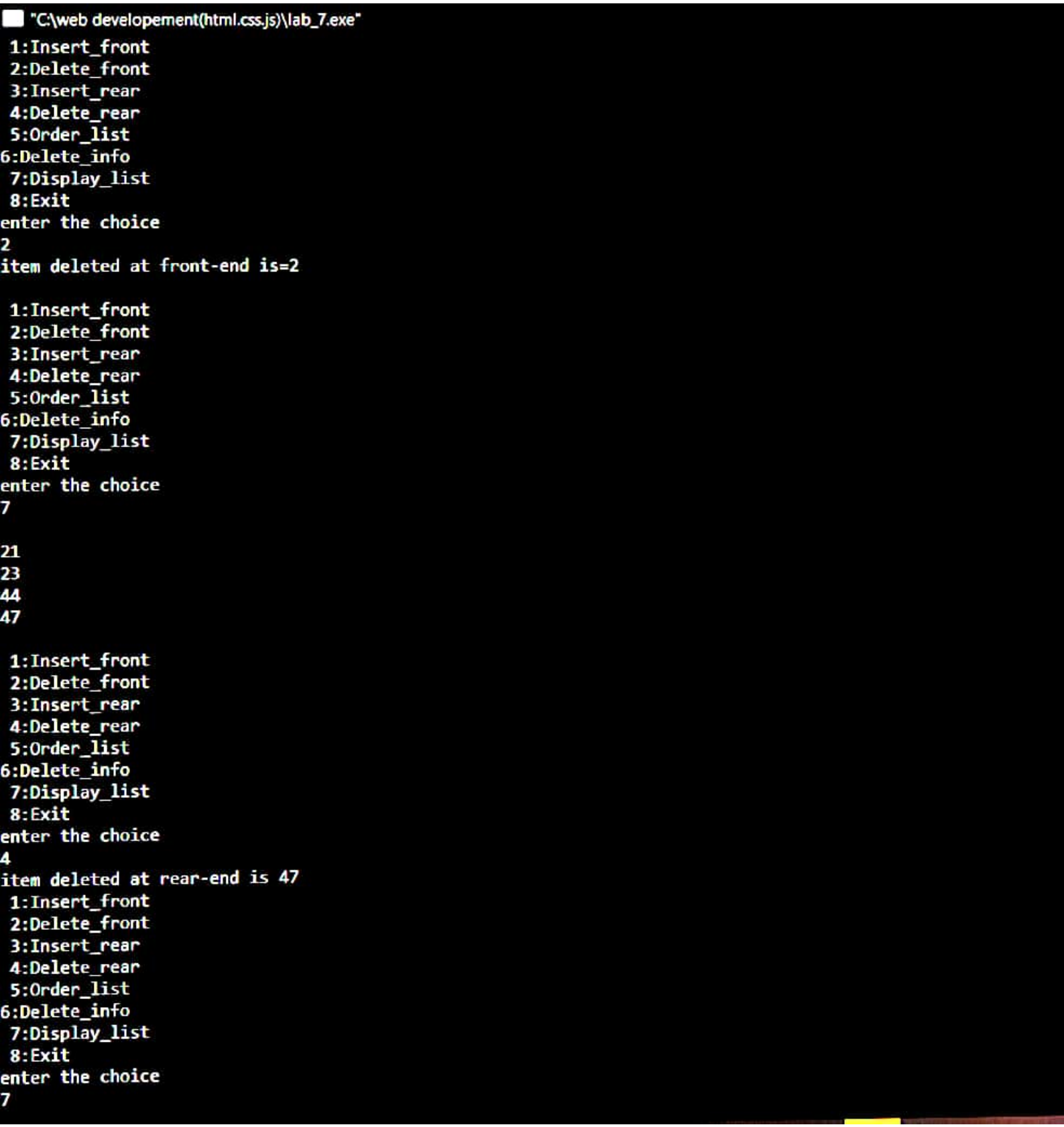
Start here | ordered_list.c | lab_8_ds.c | lab_7_ds.c | lab_5_ds.c | lab_5_6_ds.c | lab_7.c

```
133  printf("list empty cannot display items\n");
134 for(temp=first;temp!=NULL;temp=temp->link){
135  printf("%d\n",temp->info);
136 }
137 }
138 void main(){
139  int item,choice,key;
140  NODE first=NULL;
141 for(;;){
142  printf("\n 1:Insert_front\n 2:Delete_front\n 3:Insert_rear\n 4:Delete_rear\n 5:Order_list\n6:Delete_info\n 7:Display_list\n 8:Exit\n");
143  printf("enter the choice\n");
144  scanf("%d",&choice);
145 switch(choice){
146  case 1:printf("\nenter the item at front-end\n");
147  scanf("%d",&item);
148  first=insert_front(first,item);
149  break;
150  case 2:first=delete_front(first);
151  break;
152  case 3:printf("\nenter the item at rear-end\n");
153  scanf("%d",&item);
154  first=insert_rear(first,item);
155  break;
156  case 4:first=delete_rear(first);
157  break;
158  case 5:printf("\nenter the item to be inserted in ordered_list\n");
159  scanf("%d",&item);
160  first=order_list(item,first);
161  break;
162  case 6:printf("\nenter the key to be deleted\n");
163  scanf("%d",&key);
164  first=delete_info(key,first);
165  break;
166  case 7:printf("\n");
167  display(first);
168  break;
169  default:exit(0);
170  break;
171 }
172 }
173  }
174
```

C:\web developement\html,css,js\lab_7.c | C/C++ | Windows (CR+LF) | WINDOWS-1252 | Line 142, Col 92, Pos 2520 | Insert | Read/Write | default

```
"C:\web developement(html.css.js)\lab_7.exe"

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Order_list
6:Delete_info
 7:Display_list
 8:Exit
enter the choice
1

enter the item at front-end
23

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Order_list
6:Delete_info
 7:Display_list
 8:Exit
enter the choice
1

enter the item at front-end
21

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Order_list
6:Delete_info
 7:Display_list
 8:Exit
enter the choice
3

enter the item at rear-end
44

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Order_list
6:Delete_info
 7:Display_list
```





```
"C:\web developement(html.css.js)\lab_7.exe"
 7:Display_list
 8:Exit
enter the choice
7

21
23
44

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Order_list
6:Delete_info
 7:Display_list
 8:Exit
enter the choice
4
item deleted at rear-end is 44
 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Order_list
6:Delete_info
 7:Display_list
 8:Exit
enter the choice
4
item deleted at rear-end is 23
 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Order_list
6:Delete_info
 7:Display_list
 8:Exit
enter the choice
4
item deleted is 21

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Order_list
6:Delete_info
 7:Display_list
```

```
"C:\web developement(html.css.js)\lab_7.exe"
6:Delete_info
 7:Display_list
 8:Exit
enter the choice
4
list is empty cannot delete

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Order_list
6:Delete_info
 7:Display_list
 8:Exit
enter the choice
7

list empty cannot display items

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Order_list
6:Delete_info
 7:Display_list
 8:Exit
enter the choice
5

enter the item to be inserted in ordered_list
2

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Order_list
6:Delete_info
 7:Display_list
 8:Exit
enter the choice
5

enter the item to be inserted in ordered_list
45

 1:Insert_front
 2:Delete_front
```

```
"C:\web developement(html.css.js)\lab_7.exe"
enter the item to be inserted in ordered_list
45

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Order_list
6:Delete_info
 7:Display_list
 8:Exit
enter the choice
5

enter the item to be inserted in ordered_list
6

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Order_list
6:Delete_info
 7:Display_list
 8:Exit
enter the choice
7

2
6
45

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Order_list
6:Delete_info
 7:Display_list
 8:Exit
enter the choice
6

enter the key to be deleted
6
key deleted is 6
 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
```

```
"C:\web developement(html.css.js)\lab_7.exe"
 8:Exit
enter the choice
7

2
6
45

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Order_list
6:Delete_info
 7:Display_list
 8:Exit
enter the choice
6

enter the key to be deleted
6
key deleted is 6
 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Order_list
6:Delete_info
 7:Display_list
 8:Exit
enter the choice
7

2
45

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Order_list
6:Delete_info
 7:Display_list
 8:Exit
enter the choice
8

Process returned 0 (0x0)   execution time : 603.020 s
Press any key to continue.
```