

LAB - 7

```
/*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;
}
NODE insert_rear(NODE first,int item){
    NODE temp,cur;
    temp=getnode();
    temp->info=item;
    temp->link=NULL;
    if(first==NULL)
        return temp;
    cur=first;
    while(cur->link!=NULL)
        cur=cur->link;
    cur->link=temp;
    return first;
}
NODE delete_rear(NODE first){
    NODE cur,prev;
    if(first==NULL){
        printf("list is empty cannot delete\n");
        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=cur;
        cur=cur->link;
    }
    printf("item deleted at rear-end is %d",cur->info);
    free(cur);
    prev->link=NULL;
    return first;
}
NODE order_list(int item,NODE first){
    NODE temp,prev,cur;
    temp=getnode();
    temp->info=item;
    temp->link=NULL;
    if(first==NULL)

```

```

    return temp;
if(item<first->info){
    temp->link=first;
    return temp;
}
prev=NULL;
cur=first;
while(cur!=NULL&&item>cur->info){
    prev=cur;
    cur=cur->link;
}
prev->link=temp;
temp->link=cur;
return first;
}
NODE delete_info(int key,NODE first){
    NODE prev,cur;
    if(first==NULL){
        printf("list is empty\n");
        return NULL;
    }
    if(key==first->info){
        cur=first;
        first=first->link;
        freenode(cur);
        return first;
    }
    prev=NULL;
    cur=first;
    while(cur!=NULL){
        if(key==cur->info)
            break;
        prev=cur;
        cur=cur->link;
    }
    if(cur==NULL){
        printf("search is unsuccessful\n");
        return first;
    }
    prev->link=cur->link;
    printf("key deleted is %d",cur->info);
    freenode(cur);
    return first;
}

```

```

void display(NODE first){
    NODE temp;
    if(first==NULL)
        printf("list empty cannot display items\n");
    for(temp=first;temp!=NULL;temp=temp->link){
        printf("%d\n",temp->info);
    }
}

void main(){
    int item,choice,key;
    NODE first=NULL;
    for(;;){
        printf("\n 1:Insert_front\n 2:Delete_front\n 3:Insert_rear\n 4:Delete_rear\n 5:Order_list\n
6:Delete_info\n 7:Display_list\n 8:Exit\n");
        printf("enter the choice\n");
        scanf("%d",&choice);
        switch(choice){
            case 1:printf("\nenter the item at front-end\n");
                    scanf("%d",&item);
                    first=insert_front(first,item);
                    break;
            case 2:first=delete_front(first);
                    break;
            case 3:printf("\nenter the item at rear-end\n");
                    scanf("%d",&item);
                    first=insert_rear(first,item);
                    break;
            case 4:first=delete_rear(first);
                    break;
            case 5:printf("\nenter the item to be inserted in ordered_list\n");
                    scanf("%d",&item);
                    first=order_list(item,first);
                    break;
            case 6:printf("\nenter the key to be deleted\n");
                    scanf("%d",&key);
                    first=delete_info(key,first);
                    break;
            case 7:printf("\n");
                    display(first);
                    break;
            default:exit(0);
                    break;
        }
    }
}

```

}

C:\ Command Prompt

D:\coding_files\DS lab>ordered-list_linked-list

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

12

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

13

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

Command Prompt

enter the item at rear-end

33

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

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

1

enter the item at front-end

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

7

C:\ Command Prompt

```
8:Exit
enter the choice
7

2
13
12
33
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
2
item deleted at front-end is=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
7

13
12
33
45

1:Insert_front
2:Delete_front
3:Insert_rear
4:Delete_rear
5:Order_list
```


C:\> Command Prompt

```
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 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

7

13

12

33

```
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 33

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

C:\ Command Prompt

4
item deleted at rear-end is 33

- 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 12

- 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 13

- 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
list is empty cannot delete

- 1:Insert_front
- 2>Delete_front
- 3:Insert_rear
- 4>Delete_rear
- 5:Order_list
- 6>Delete_info

C:\ Command Prompt

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

34

CA. Command Prompt

enter the item to be inserted in ordered_list

34

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

78

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

4

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

C:\ Command Prompt

enter the item to be inserted in ordered_list

1

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

1

2

4

34

78

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

2

key deleted is 2

1:Insert_front

2:Delete_front

3:Insert_rear

4:Delete_rear

5:Order_list

6:Delete_info

7:Display_list

8:Exit

C:\ Command Prompt

6

enter the key to be deleted

2

key deleted is 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

7

1

4

34

78

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

D:\coding files\DS lab>