

```

1  #include<stdio.h>
2  #include<process.h>
3  struct node
4  {
5      int info;
6      struct node *link;
7  };
8  typedef struct node *NODE;
9  NODE getnode()
10 {
11     NODE x;
12     x=(NODE)malloc(sizeof(struct node));
13     if(x==NULL)
14     {
15         printf("Memory is full\n");
16         exit(0);
17     }
18     return x;
19 }
20 void freenode(NODE x)
21 {
22     free(x);
23 }
24 NODE insert_front(NODE first,int item)
25 {
26     NODE temp;
27     temp=getnode();
28     temp->info=item;
29     temp->link=NULL;
30     if(first==NULL)
31         return temp;
32     temp->link=first;
33     first=temp;
34     return first;

```

```

35     }
36     NODE delete_front(NODE first)
37     {
38         NODE temp;
39         if(first==NULL)
40         {
41             printf("CANNOT DELETE AS LIST IS EMPTY\n");
42             return first;
43         }
44         temp=first;
45         temp=temp->link;
46         printf("ITEM DELETED AT FRONT END=%d\n",first->info);
47         free(first);
48         return temp;
49     }
50     NODE delete_rear(NODE first)
51     {
52         NODE cur,prev;
53         if(first==NULL)
54         {
55             printf("CANNOT DELETE AS LIST IS EMPTY\n");
56             return first;
57         }
58         if(first->link==NULL)
59         {
60             printf("ITEM DELETED=%d\n",first->info);
61             free(first);
62             return NULL;
63         }
64         prev=NULL;
65         cur=first;
66         while(cur->link!=NULL)
67         {
68             prev=cur;

```

```

69     cur=cur->link;
70 }
71 printf("ITEM DELETED AT REAR END=%d\n",cur->info);
72 free(cur);
73 prev->link=NULL;
74 return first;
75 }
76 void display(NODE first)
77 {
78     NODE temp;
79     if(first==NULL)
80         printf("list empty cannot display items\n");
81     for(temp=first;temp!=NULL;temp=temp->link)
82     {
83         printf("%d\n",temp->info);
84     }
85 }
86 void main()
87 {
88     int item,choice,pos;
89     NODE first=NULL,first1=NULL;
90     for(;;)
91     {
92         printf("1.Push in Stack\n2.Insert in Queue\n3.Pop from Stack\n4.Delete from rear in Queue\n5.Display Stack\n6.Display Queue\n7.Exit\n");
93         printf("enter the choice\n");
94         scanf("%d",&choice);
95         switch(choice)
96         {
97             case 1:printf("Enter item to be pushed in Stack:\n");
98                     scanf("%d",&item);
99                     first=insert_front(first,item);
100                    break;
101             case 2:printf("Enter item to be inserted in Queue:\n");
102                     scanf("%d",&item);

```

```
95     switch(choice)
96     {
97         case 1:printf("Enter item to be pushed in Stack:\n");
98                 scanf("%d",&item);
99                 first=insert_front(first,item);
100                break;
101         case 2:printf("Enter item to be inserted in Queue:\n");
102                 scanf("%d",&item);
103                 first1=insert_front(first1,item);
104                break;
105         case 3:first=delete_front(first);
106                break;
107         case 4:first1=delete_rear(first1);
108                break;
109         case 5:display(first);
110                break;
111         case 6:display(first1);
112                break;
113         default:exit(0);
114                break;
115     }
116 }
117 }
118 }
```

```
1.Push in Stack
2.Insert in Queue
3.Pop from Stack
4.Delete from rear in Queue
5.Display Stack
6.Display Queue
7.Exit
enter the choice
```

```
1
Enter item to be pushed in Stack:
89
```

```
1.Push in Stack
2.Insert in Queue
3.Pop from Stack
4.Delete from rear in Queue
5.Display Stack
6.Display Queue
7.Exit
enter the choice
```

```
1
Enter item to be pushed in Stack:
23
```

```
1.Push in Stack
2.Insert in Queue
3.Pop from Stack
4.Delete from rear in Queue
5.Display Stack
6.Display Queue
7.Exit
enter the choice
```

```
1
Enter item to be pushed in Stack:
45
```

```
1.Push in Stack
2.Insert in Queue
3.Pop from Stack
4.Delete from rear in Queue
5.Display Stack
6.Display Queue
7.Exit
enter the choice
```

```
5
45
23
89
```

```
1.Push in Stack
2.Insert in Queue
3.Pop from Stack
4.Delete from rear in Queue
5.Display Stack
```



```
5.Display Stack
6.Display Queue
7.Exit
enter the choice
3
ITEM DELETED AT FRONT END=89
1.Push in Stack
2.Insert in Queue
3.Pop from Stack
4.Delete from rear in Queue
5.Display Stack
6.Display Queue
7.Exit
enter the choice
3
CANNOT DELETE AS LIST IS EMPTY
1.Push in Stack
2.Insert in Queue
3.Pop from Stack
4.Delete from rear in Queue
5.Display Stack
6.Display Queue
7.Exit
enter the choice
2
Enter item to be inserted in Queue:
66
1.Push in Stack
2.Insert in Queue
3.Pop from Stack
4.Delete from rear in Queue
5.Display Stack
6.Display Queue
7.Exit
enter the choice
2
Enter item to be inserted in Queue:
33
1.Push in Stack
2.Insert in Queue
3.Pop from Stack
4.Delete from rear in Queue
5.Display Stack
6.Display Queue
7.Exit
enter the choice
2
Enter item to be inserted in Queue:
```

Enter item to be inserted in Queue:

55

1.Push in Stack

2.Insert in Queue

3.Pop from Stack

4.Delete from rear in Queue

5.Display Stack

6.Display Queue

7.Exit

enter the choice

6

55

33

66

1.Push in Stack

2.Insert in Queue

3.Pop from Stack

4.Delete from rear in Queue

5.Display Stack

6.Display Queue

7.Exit

enter the choice

4

ITEM DELETED AT REAR END=66

1.Push in Stack

2.Insert in Queue

3.Pop from Stack

4.Delete from rear in Queue

5.Display Stack

6.Display Queue

7.Exit

enter the choice

4

ITEM DELETED AT REAR END=33

1.Push in Stack

2.Insert in Queue

3.Pop from Stack

4.Delete from rear in Queue

5.Display Stack

6.Display Queue

7.Exit

enter the choice

4

ITEM DELETED=55

1.Push in Stack

2.Insert in Queue

3.Pop from Stack

4.Delete from rear in Queue

```
1.Push in Stack
2.Insert in Queue
3.Pop from Stack
4.Delete from rear in Queue
5.Display Stack
6.Display Queue
7.Exit
```

enter the choice

4

CANNOT DELETE AS LIST IS EMPTY

```
1.Push in Stack
2.Insert in Queue
3.Pop from Stack
4.Delete from rear in Queue
5.Display Stack
6.Display Queue
7.Exit
```

enter the choice

7

Process returned 0 (0x0) execution time : 84.160 s

Press any key to continue.

—