```
#include<stdio.h>
     #include<stdlib.h>
     struct node
    □ {
       int info:
       struct node *link;
 8
     typedef struct node *NODE;
     NODE getnode()
    ₽{
11
     NODE X;
     x=(NODE) malloc(sizeof(struct node));
13
     if (x==NULL)
14
15
16
       printf("Memory full\n");
17
       exit(0);
18
19
      return x;
20
     void freenode (NODE x)
    □ {
    free(x);
     NODE insert front (NODE first, int item)
26
    ₽{
     NODE temp;
     temp=getnode();
     temp->info=item;
30
     temp->link=NULL;
31
     if(first==NULL)
     return temp;
33
     temp->link=first;
34
     first=temp;
35
    return first;
36
37
38
     NODE insert rear (NODE first, int item)
```

```
NODE insert rear (NODE first, int item)
    □ {
    NODE temp, cur;
41
    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;
51
52
     NODE delete front (NODE first)
    NODE temp;
    if (first==NULL)
57
58
    printf("list is empty cannot delete\n");
    return first;
61
    temp=first;
    temp=temp->link;
    printf("item deleted at front-end is=%d\n", first->info);
    free (first);
    return temp;
67
     NODE delete rear (NODE first)
    ₽ {
69
    NODE cur, prev;
    if (first==NULL)
    printf("List is empty cannot delete\n");
    return first;
```

```
free (first);
      return temp;
 67
      NODE delete rear (NODE first)
 69
     日{
      NODE cur, prev;
      if (first==NULL)
 71
 72
 73
      printf("List is empty cannot delete\n");
 74
      return first;
 75
 76
      if (first->link==NULL)
 77
 78
      printf("Item deleted is %d\n", first->info);
 79
      free (first);
      return NULL;
 81
      prev=NULL;
      cur=first;
 83
      while (cur->link!=NULL)
 84
 85
     白(
 86
      prev=cur;
      cur=cur->link;
 88
      printf("Item deleted at rear-end is %d", cur->info);
 89
      free (cur);
      prev->link=NULL;
      return first;
 93
 94
 95
      void display (NODE first)
 96
 97
     ₽{
 98
       NODE temp;
 99
       if (first==NULL)
100
101
       printf("List empty cannot display items\n");
102
       return;
<
```

```
void display (NODE first)
     ₽{
       NODE temp;
       if (first==NULL)
100
101
       printf("List empty cannot display items\n");
102
       return:
103
       printf("Contents of list:\n");
104
       for(temp=first;temp!=NULL;temp=temp->link)
105
106
        printf("%d\n", temp->info);
107
108
109
110
111
      int main()
112
113
      int item, choice;
      NODE first=NULL, a, b;
114
115
116
     for(;;)
117
      printf("\n1:Stack\n2:Queue\n3:Exit\n");
118
      printf("Enter the choice\n");
119
      scanf ("%d", &choice);
120
      switch (choice)
121
122
123
124
        case 1:printf("Stack\n");
125
            for(;;)
126
              printf("\n 1:Insert rear\n 2:Delete rear\n 3:Display list\n 4:Exit\n");
127
              printf("Enter the choice\n");
128
               scanf ("%d", &choice);
129
               switch (choice)
130
131
               case 1:printf("Enter the item at rear-end\n");
132
                   scanf ("%d", &item);
133
```

```
131
               case 1:printf("Enter the item at rear-end\n");
132
                   scanf("%d", &item);
133
                   first=insert rear(first,item);
134
                   break;
135
              case 2:first=delete rear(first);
136
                   break:
137
               case 3:display(first);
138
139
                   break;
              default:exit(0);
140
141
                  break;
142
143
144
        case 2:printf("OUEUE\n");
145
              for(;;)
146
                   printf("\n 1:Insert rear\n 2:Delete front\n 3:Display list\n 4:Exit\n");
147
                   printf("Enter the choice\n");
148
                   scanf ("%d", &choice);
149
150
                   switch (choice)
151
                   case 1:printf("Enter the item at rear-end\n");
152
153
                           scanf("%d", &item);
154
                           first=insert rear(first,item);
                           break;
155
                   case 2:first=delete front(first);
156
157
                           break;
158
                   case 3:display(first);
159
                           break;
160
                   default:exit(0);
                           break;
161
162
163
164
165
       case 3:exit(0);
        default:printf("Invalid choice\n");
166
167
168
```

```
135
                   break;
              case 2:first=delete rear(first);
136
                  break;
137
138
              case 3:display(first);
139
                  break;
              default:exit(0);
140
141
                  break;
142
143
144
        case 2:printf("QUEUE\n");
145
              for(;;)
146
                   printf("\n 1:Insert rear\n 2:Delete front\n 3:Display list\n 4:Exit\n");
147
                   printf("Enter the choice\n");
148
                   scanf ("%d", &choice);
149
150
                   switch (choice)
151
                   case 1:printf("Enter the item at rear-end\n");
152
                           scanf ("%d", &item);
153
                           first=insert rear(first,item);
154
155
                           break;
                   case 2:first=delete front(first);
156
                           break;
157
                   case 3:display(first);
158
159
                           break;
160
                   default:exit(0);
161
                           break;
162
163
164
       case 3:exit(0);
165
        default:printf("Invalid choice\n");
166
167
168
      return 0;
169
170
171
```

first=insert rear(first,item);

```
1:Stack
2:Queue
3:Exit
Enter the choice
1
Stack
1: Insert_rear
2:Delete_rear
3:Display_list
4:Exit
Enter the choice
Enter the item at rear-end
1
1:Insert_rear
2:Delete_rear
3:Display_list
4:Exit
Enter the choice
1
Enter the item at rear-end
1:Insert_rear
2:Delete_rear
3:Display_list
4:Exit
Enter the choice
1
Enter the item at rear-end
3
1:Insert_rear
2:Delete_rear
3:Display_list
4:Exit
Enter the choice
3
Contents of list:
```

## C:\WINDOWS\SYSTEM32\cmd.exe Enter the choice 3 Contents of list: 1 2 3 1:Insert\_rear 2:Delete\_rear 3:Display\_list 4:Exit Enter the choice 2 Item deleted at rear-end is 3 1:Insert\_rear 2:Delete\_rear 3:Display list 4:Exit Enter the choice 2 Item deleted at rear-end is 2 1:Insert\_rear 2:Delete\_rear 3:Display\_list 4:Exit Enter the choice 2 Item deleted is 1 1:Insert\_rear 2:Delete\_rear 3:Display\_list

4:Exit

4:Exit

3

Enter the choice

1:Insert\_rear 2:Delete\_rear 3:Display\_list

Enter the choice

List empty cannot display items

```
C:\WINDOWS\SYSTEM32\cmd.exe
1:Insert rear
2:Delete_rear
3:Display_list
4:Exit
Enter the choice
2
Item deleted at rear-end is 3
1:Insert rear
2:Delete_rear
3:Display_list
4:Exit
Enter the choice
2
Item deleted at rear-end is 2
1:Insert rear
2:Delete_rear
3:Display_list
4:Exit
Enter the choice
2
Item deleted is 1
1:Insert_rear
2:Delete rear
3:Display_list
4:Exit
Enter the choice
3
List empty cannot display items
1:Insert_rear
2:Delete_rear
3:Display_list
4:Exit
Enter the choice
```

(program exited with code: 0)

```
1:Stack
2:Queue
3:Exit
Enter the choice
2
QUEUE
1: Insert_rear
2:Delete_front
3:Display_list
4:Exit
Enter the choice
Enter the item at rear-end
1
1: Insert_rear
2:Delete_front
3:Display_list
4:Exit
Enter the choice
Enter the item at rear-end
1:Insert_rear
2:Delete_front
3:Display_list
4:Exit
Enter the choice
1
Enter the item at rear-end
3
1:Insert_rear
2:Delete_front
3:Display_list
4:Exit
Enter the choice
```

Contents of list:

## C:\WINDOWS\SYSTEM32\cmd.exe Enter the choice 3 Contents of list: 1 2

item deleted at front-end is=1

item deleted at front-end is=2

item deleted at front-end is=3

List empty cannot display items

1:Insert\_rear 2:Delete\_front 3:Display\_list

Enter the choice

1:Insert\_rear 2:Delete\_front 3:Display\_list

4:Exit

4:Exit

4:Exit

4:Exit

3

```
C:\WINDOWS\SYSTEM32\cmd.exe
1: Insert_rear
2:Delete_front
3:Display_list
4:Exit
Enter the choice
item deleted at front-end is=2
1: Insert_rear
2:Delete front
3:Display_list
4:Exit
Enter the choice
2
item deleted at front-end is=3
1:Insert rear
2:Delete_front
3:Display_list
4:Exit
Enter the choice
List empty cannot display items
1:Insert_rear
```

2:Delete\_front
3:Display\_list

Enter the choice

(program exited with code: 0)

Press any key to continue . . .

4:Exit