```
#include<malloc.h>
     #include<stdlib.h>
     struct node
 5
   ⊟ {
      int info:
      struct node *link;
     typedef struct node *NODE;
     NODE getnode()
11
12
    NODE X:
     x=(NODE) malloc(sizeof(struct node));
14
     if (x==NULL)
15
      printf("Memory full\n");
16
      exit(0);
17
19
      return x;
     void freenode (NODE x)
    free(x);
     NODE insert rear (NODE first, int item)
     NODE temp, cur;
     temp=getnode();
30
31
     temp->info=item;
     temp->link=NULL;
33
    if (first==NULL)
34
     return temp;
35
     cur=first;
     while (cur->link!=NULL)
36
37
      cur=cur->link;
     cur->link=temp;
38
```

#include<stdio.h>

```
38
    cur->link=temp;
39
    return first;
40
41
42
     NODE delete rear (NODE first)
44
    ₽ {
    NODE cur, prev;
    if (first==NULL)
47
     printf("List is empty cannot delete\n");
48
     return first;
49
50
    if (first->link==NULL)
51
52
     printf("Item deleted is %d\n", first->info);
     free (first);
54
     return NULL;
56
57
     prev=NULL;
     cur=first;
58
     while (cur->link!=NULL)
60
61
     prev=cur;
     cur=cur->link;
63
    printf("Item deleted at rear-end is %d", cur->info);
64
    free (cur);
65
     prev->link=NULL;
    return first;
67
68
69
70
71
     NODE insert pos(int item, int pos, NODE first)
72
    日{
    NODE temp, cur, prev;
74
    int count;
    temp=getnode();
```

```
75
      temp=getnode();
 76
      temp->info=item;
 77
      temp->link=NULL;
 78
      if (first==NULL&&pos==1)
 79
 80
      return temp;
 81
 82
      if (first==NULL)
 83
      printf("Invalid position\n");
 84
 85
      return first:
 86
 87
      if (pos==1)
 88
 89
      temp->link=first;
      first=temp;
 90
 91
      return temp;
 92
 93
      count=1;
 94
      prev=NULL;
 95
      cur=first;
 96
      while (cur!=NULL&&count!=pos)
 97
 98
      prev=cur;
 99
      cur=cur->link;
100
      count++;
101
      if (count==pos)
102
103
      prev->link=temp;
104
      temp->link=cur;
105
      return first;
106
107
      printf("Invalid position\n");
108
      return first;
109
110
111
```

int count;

74

<

```
112
113
      NODE delete pos(int pos, NODE first)
114
     ₽ {
115
      NODE cur;
116
      NODE prev;
117
      int count, flag=0;
118
      if(first==NULL || pos<0)
119
120
      printf("Invalid position\n");
121
      return NULL;
122
123
      if (pos==1)
124
125
      cur=first;
126
     first=first->link;
127
      freenode (cur);
128
      return first;
129
130
      prev=NULL;
      cur=first;
131
132
      count=1;
133
      while (cur!=NULL)
134
     白(
135
      if (count==pos) {flag=1;break;}
136
      count++;
137
      prev=cur;
138
      cur=cur->link;
139
140
      if(flag==0)
141
142
      printf("Invalid position\n");
      return first;
143
144
      printf("Item deleted at given position is %d\n", cur->info);
145
146
      prev->link=cur->link;
147
      freenode (cur);
148
      return first;
```

111

<

```
153
     ₽{
       NODE temp;
154
155
       if (first==NULL)
156
       printf("List is empty cannot display items\n");
       for(temp=first; temp!=NULL; temp=temp->link)
157
158
159
       printf("%d\n", temp->info);
160
161
162
163
      int main()
164
165
     目{
      int item, choice, pos;
166
      NODE first=NULL;
167
168
      for(;;)
169
170
      printf("\n 1:Insert rear\n 2:Delete rear\n");
      printf(" 3:Insert info position\n 4:Delete info position\n 5:Display list\n 6:Exit\n");
171
      printf("Enter the choice : ");
172
      scanf("%d", &choice);
173
174
      switch (choice)
175
       case 1:printf("Enter the item at rear-end:\n");
176
177
       scanf ("%d", &item);
       first=insert rear(first,item);
178
       break:
179
180
       case 2:first=delete rear(first);
181
       break;
       case 3:printf("Enter the item to be inserted at given position:\n");
182
183
       scanf ("%d", &item);
       printf("Enter the position:\n");
184
185
       scanf ("%d", &pos);
186
       first=insert pos(item, pos, first);
187
       break;
       case 4:printf("Enter the position:\n");
188
189
       scanf ("%d", &pos);
<
```

void display (NODE first)

152

```
164
      int main()
165
    E!
      int item, choice, pos;
166
167
      NODE first=NULL;
168
      for(;;)
169
170
      printf("\n 1:Insert rear\n 2:Delete rear\n");
      printf(" 3:Insert info position\n 4:Delete info position\n 5:Display list\n 6:Exit\n");
171
172
      printf("Enter the choice : ");
173
      scanf("%d", &choice);
174
      switch (choice)
175
176
       case 1:printf("Enter the item at rear-end:\n");
177
       scanf("%d", &item);
178
       first=insert rear(first,item);
179
       break;
       case 2:first=delete rear(first);
180
181
       break:
       case 3:printf("Enter the item to be inserted at given position:\n");
182
       scanf("%d", &item);
183
       printf("Enter the position:\n");
184
185
       scanf ("%d", &pos);
       first=insert pos(item, pos, first);
186
       break;
187
       case 4:printf("Enter the position:\n");
188
189
       scanf ("%d", &pos);
       first=delete pos(pos, first);
190
191
       break:
192
       case 5:display(first);
193
       break:
       default:exit(0);
194
195
       break;
196
197
198
      return 0:
199
200
```

163

<

```
1: Insert rear
2:Delete_rear
3:Insert_info_position
4:Delete_info_position
5:Display list
6:Exit
Enter the choice : 1
Enter the item at rear-end:
1:Insert_rear
2:Delete_rear
3:Insert_info_position
4:Delete_info_position
5:Display list
6:Exit
Enter the choice : 1
Enter the item at rear-end:
2
1: Insert_rear
2:Delete_rear
3:Insert_info_position
4:Delete_info_position
5:Display_list
6:Exit
Enter the choice : 1
Enter the item at rear-end:
3
1:Insert_rear
2:Delete_rear
3:Insert_info_position
4:Delete info position
5:Display_list
6:Exit
Enter the choice : 3
Enter the item to be inserted at given position:
12
Enter the position:
```

```
C:\WINDOWS\SYSTEM32\cmd.exe
12
Enter the position:
2
1: Insert_rear
2:Delete rear
3:Insert_info_position
4:Delete info position
5:Display_list
6:Exit
Enter the choice : 3
Enter the item to be inserted at given position:
10
Enter the position:
3
1:Insert_rear
2:Delete_rear
3:Insert_info_position
4:Delete_info_position
5:Display list
6:Exit
Enter the choice : 5
1
12
10
2
3
1:Insert_rear
2:Delete_rear
3:Insert_info_position
4:Delete_info_position
5:Display_list
6:Exit
Enter the choice : 4
Enter the position:
Item deleted at given position is 10
1:Insert_rear
```

2:Delete_rear

```
1: Insert_rear
2:Delete_rear
3:Insert_info_position
4:Delete info position
5:Display_list
6:Exit
Enter the choice : 4
Enter the position:
1:Insert_rear
2:Delete_rear
3:Insert_info_position
4:Delete_info_position
5:Display_list
6:Exit
Enter the choice: 4
Enter the position:
Invalid position
1: Insert_rear
2:Delete_rear
3:Insert_info_position
4:Delete_info_position
5:Display_list
6:Exit
Enter the choice : 5
12
2
1: Insert_rear
2:Delete_rear
3:Insert_info_position
4:Delete_info_position
5:Display_list
6:Exit
Enter the choice : 4
Enter the position:
1
```

```
C:\WINDOWS\SYSTEM32\cmd.exe
3:Insert_info_position
4:Delete_info_position
5:Display_list
6:Exit
Enter the choice : 5
12
2
3
1: Insert_rear
2:Delete_rear
3: Insert_info_position
4:Delete_info_position
5:Display list
6:Exit
Enter the choice : 4
Enter the position:
1: Insert rear
2:Delete rear
3:Insert_info_position
4:Delete_info_position
5:Display list
6:Exit
Enter the choice : 4
Enter the position:
Item deleted at given position is 3
1:Insert rear
2:Delete rear
3:Insert_info_position
4:Delete_info_position
5:Display_list
```

6:Exit

Enter the choice : 4 Enter the position:

Invalid position

1:Insert_rear

```
C:\WINDOWS\SYSTEM32\cmd.exe
4:Delete_info_position
5:Display_list
6:Exit
Enter the choice: 4
Enter the position:
1
1: Insert_rear
2:Delete rear
3:Insert_info_position
4:Delete_info_position
5:Display_list
6:Exit
Enter the choice : 4
Enter the position:
Item deleted at given position is 3
1: Insert rear
2:Delete rear
3:Insert_info_position
4:Delete_info_position
5:Display_list
6:Exit
Enter the choice: 4
Enter the position:
```

Invalid position

3:Insert_info_position
4:Delete_info_position

Enter the choice: 45

(program exited with code: 0)

Press any key to continue

1:Insert_rear 2:Delete_rear

5:Display_list

6:Exit