```
1: #include <stdio.h>
 2: #include <stdlib.h>
 3:
 4: struct Node *f = NULL;
 5: struct Node *r = NULL;
 6:
 7: struct Node
8: {
 9:
        int data;
        struct Node *next;
10:
11: };
12:
13: void linkedListTraversal(struct Node *ptr)
14: {
15:
        printf("Printing the elements of this linked list\n");
        while (ptr != NULL)
16:
17:
        {
18:
            printf("Element: %d\n", ptr->data);
            ptr = ptr->next;
19:
20:
        }
21: }
22:
23: void enqueue(int val)
24: {
25:
        struct Node *n = (struct Node *) malloc(sizeof(struct Node));
26:
        if(n==NULL){
27:
            printf("Queue is Full");
28:
29:
        else{
            n->data = val;
30:
31:
            n->next = NULL;
            if(f==NULL){
32:
33:
                 f=r=n;
34:
35:
            else{
36:
                 r-next = n;
37:
                 r=n;
38:
            }
39:
        }
```

```
40: }
41:
42: int dequeue()
43: {
44:
        int val = -1;
        struct Node *ptr = f;
45:
        if(f==NULL){
46:
            printf("Queue is Empty\n");
47:
48:
        }
49:
        else{
            f = f->next;
50:
            val = ptr->data;
51:
            free(ptr);
52:
53:
54:
        return val;
55: }
56:
57: int main()
58: {
59:
        linkedListTraversal(f);
        printf("Dequeuing element %d\n", dequeue());
60:
        enqueue(34);
61:
        enqueue(4);
62:
63:
        enqueue(7);
64:
        enqueue(17);
        printf("Dequeuing element %d\n", dequeue());
65:
        printf("Dequeuing element %d\n", dequeue());
66:
67:
        printf("Dequeuing element %d\n", dequeue());
68:
        printf("Dequeuing element %d\n", dequeue());
69:
        linkedListTraversal(f);
70:
        return 0:
71: }
72:
```