

```

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;
10:    struct Node *next;
11: };
12:
13: void linkedListTraversal(struct Node *ptr)
14: {
15:     printf("Printing the elements of this linked list\n");
16:     while (ptr != NULL)
17:     {
18:         printf("Element: %d\n", ptr->data);
19:         ptr = ptr->next;
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{
30:         n->data = val;
31:         n->next = NULL;
32:         if(f==NULL){
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;
45:     struct Node *ptr = f;
46:     if(f==NULL){
47:         printf("Queue is Empty\n");
48:     }
49:     else{
50:         f = f->next;
51:         val = ptr->data;
52:         free(ptr);
53:     }
54:     return val;
55: }
56:
57: int main()
58: {
59:     linkedListTraversal(f);
60:     printf("Dequeuing element %d\n", dequeue());
61:     enqueue(34);
62:     enqueue(4);
63:     enqueue(7);
64:     enqueue(17);
65:     printf("Dequeuing element %d\n", dequeue());
66:     printf("Dequeuing element %d\n", dequeue());
67:     printf("Dequeuing element %d\n", dequeue());
68:     printf("Dequeuing element %d\n", dequeue());
69:     linkedListTraversal(f);
70:     return 0;
71: }
72:
```