

Name - Sanjoy Saha

Stream - Computer Science &
Engineering

Sec - A

Roll no. - 3

University :- 10900120003
Roll no.

Subject - DSA Lab

C QueueusingLinkedList.c > main()

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <malloc.h>
4
5  struct node {
6      int data;
7      struct node *next;
8  };
9
10 struct node *beg;
11 int length=0, max;
12 int item;
13 int front = -1, rear=-1;
14
15 int main() {
16
17     int choice;
18
19     printf("Enter the MAXSIZE of the queue: ");
20     scanf("%d", &max);
21
22     options:
23         printf("Queue operations: \n");
24         printf("1. Insert\n");
25         printf("2. Delete\n");
26         printf("3. Display\n");
27         printf("4. EXIT\n");
28
29     printf("\nChoose an options to do the operation: ");
30     scanf("%d", &choice);
```

```
32     switch(choice) {
33     case 1:
34         Qinsert();
35         goto options;
36         break;
37     case 2:
38         Qdelete();
39         goto options;
40         break;
41     case 3:
42         Qdisplay();
43         goto options;
44         break;
45     case 4:
46         printf("Successfully Exited...");
47         break;
48     default:
49         printf("Wrong Input Provided\n");
50         goto options;
51     }
52     return 0;
53 }
54
55 int Qinsert() {
56
57     struct node *temp, *insert;
58     int value;
59
60     if ((rear-front+1) >= max) {
61         printf("Queue Overflown...\n");
```

```
60     if ((rear-front+1) >= max) {
61         printf("Queue Overflow...\n");
62         return 0;
63     }
64     printf("Enter the value to insert in queue: ");
65     scanf("%d", &value);
66
67     if (front == -1) {
68         beg = (struct node *)malloc(sizeof(struct node));
69         beg->next = NULL;
70         beg->data = value;
71     } else {
72         temp = beg;
73
74         while(temp->next != NULL) {
75             temp = temp->next;
76         }
77
78         insert = (struct node *)malloc(sizeof(struct node));
79
80         insert->data = value;
81         insert->next = NULL;
82         temp->next = insert;
83     }
84     rear += 1;
85     if (front == -1) {
86         front = 0;
87     }
88     return 0;
89 }
```

```
91 int Qdelete() {
92
93     if (front == -1) {
94         printf("Queue Underflow!\n");
95         return 0;
96     }
97
98     if (front == rear) {
99         front = -1;
100        rear = -1;
101    } else {
102        front += 1;
103    }
104
105    beg = beg->next;
106    printf("Successfully delete operation done...\n");
107
108    return 0;
109 }
110
```

```
110
111 int Qdisplay() {
112     struct node *temp;
113     int i;
114     if (front == -1) {
115         printf("Queue Empty!!\n");
116         return 0;
117     }
118
119     temp = beg;
120
121     printf("The elements of the queue are: \n");
122
123     while (temp != NULL) {
124         printf("%d ", temp->data);
125         temp = temp->next;
126     }
127
128     printf("\n");
129
130     return 0;
131 }
132
```

Enter the MAXSIZE of the queue: 5

Queue operations:

1. Insert
2. Delete
3. Display
4. EXIT

Choose an options to do the operation: 1

Enter the value to insert in queue: 21

Queue operations:

1. Insert
2. Delete
3. Display
4. EXIT

Choose an options to do the operation: 1

Enter the value to insert in queue: 223

Queue operations:

1. Insert
2. Delete
3. Display
4. EXIT

Choose an options to do the operation: 1

Enter the value to insert in queue: 6766

Queue operations:

1. Insert
2. Delete
3. Display
4. EXIT

Choose an options to do the operation: 1

Enter the value to insert in queue: 3

Queue operations:

1. Insert
2. Delete
3. Display
4. EXIT

Queue operations:

1. Insert
2. Delete
3. Display
4. EXIT

Choose an options to do the operation: 3

The elements of the queue are:

21 223 6766 3

Queue operations:

1. Insert
2. Delete
3. Display
4. EXIT

Choose an options to do the operation: 1

Enter the value to insert in queue: 409

Queue operations:

1. Insert
2. Delete
3. Display
4. EXIT

Choose an options to do the operation: 3

The elements of the queue are:

21 223 6766 3 409

Queue operations:

1. Insert
2. Delete
3. Display
4. EXIT

Choose an options to do the operation: 2

Successfully delete operation done...

Queue operations:

1. Insert
2. Delete
3. Display
4. EXIT

Choose an options to do the operation: 2

Successfully delete operation done...

Queue operations:

1. Insert
2. Delete

4. EXIT

Choose an options to do the operation: 1

Enter the value to insert in queue: 409

Queue operations:

1. Insert
2. Delete
3. Display
4. EXIT

Choose an options to do the operation: 3

The elements of the queue are:

21 223 6766 3 409

Queue operations:

1. Insert
2. Delete
3. Display
4. EXIT

Choose an options to do the operation: 2

Successfully delete operation done...

Queue operations:

1. Insert
2. Delete
3. Display
4. EXIT

Choose an options to do the operation: 2

Successfully delete operation done...

Queue operations:

1. Insert
2. Delete
3. Display
4. EXIT

Choose an options to do the operation: 3

The elements of the queue are:

6766 3 409

Queue operations:

1. Insert
2. Delete
3. Display
4. EXIT

Choose an options to do the operation: