

## Experiments/linearQueue.c

```
1  #include <stdio.h>
2  #define max 100
3
4  int queue[max];
5  int front = -1, rear = -1;
6
7  void insert(void);
8  int delete(void);
9  int peek(void);
10 int display(void);
11
12 int main()
13 {
14     int op;
15
16     do{
17         printf("\nMenu");
18         printf("\n1.Insert\n2.Delete\n3.Peek\n4.Display");
19         printf("\nEnter your choice: ");
20         scanf("%d",&op);
21
22         switch(op){
23             case 1:
24                 insert();
25                 break;
26
27             case 2:
28                 delete();
29                 break;
30
31             case 3:
32                 peek();
33                 break;
34
35             case 4:
36                 display();
37                 break;
38
39             default:
40                 printf("\nInvalid Input\n");
41         }
42     }
43     while(op != 5);
44
45     return 0;
46 }
47
48 void insert(){
49     int val;
50
51     if(rear == max - 1){
```

```
52     printf("\nOverflow\n");
53 }
54 else if(front == -1 && rear == -1){
55     printf("\nEnter the element: ");
56     scanf("%d",&val);
57     front = 0;
58     queue[++rear] = val;
59 }
60 else{
61     printf("\nEnter the element: ");
62     scanf("%d",&val);
63     queue[++rear] = val;
64 }
65 }
66
67 int delete(){
68     int num;
69
70     if(front == -1 || front > rear){
71         printf("\nUnderflow\n");
72         return -1;
73     }
74     else{
75         num = queue[front];
76         front++;
77         printf("\nDeleted element: %d\n",num);
78     }
79     if(front > rear){
80         front = rear = -1;
81     }
82 }
83
84 int peek(){
85     if(front == -1 || front > rear){
86         printf("\nQueue is empty\n");
87         return -1;
88     }
89     else{
90         printf("\n%d\n",queue[front]);
91     }
92 }
93
94 int display(){
95     if(front == -1 && rear == -1){
96         printf("\nQueue is empty\n");
97         return -1;
98     }
99     for(int i=front; i<=rear; i++){
100         printf("\n%d",queue[i]);
101     }
102     printf("\n");
103 }
104
105 /*
```

```
106 cd "/home/gagan/Desktop/Data_Structure_And_Algorithm/Experiments/" && gcc
    linearQueue.c -o linearQueue && "/home/gagan/Desktop/Data_Structure_And_Al
lgogagan@computer-ThinkCentre:~/Desktop/Data_Structure_And_Algorithm$ cd
"/home/gagan/Desktop/Data_Structure_And_Algorithm/Experiments/" && gcc linearQueue.c
-o linearQueue && "/home/gagan/Desktop/Data_Structure_And_Al
lgorithm/Experiments/"linearQueue
107
108 Menu
109 1.Insert
110 2.Delete
111 3.Peek
112 4.Display
113 Enter your choice: 1
114
115 Enter the element: 1
116
117 Menu
118 1.Insert
119 2.Delete
120 3.Peek
121 4.Display
122 Enter your choice: 1
123
124 Enter the element: 45
125
126 Menu
127 1.Insert
128 2.Delete
129 3.Peek
130 4.Display
131 Enter your choice: 1
132
133 Enter the element: 78
134
135 Menu
136 1.Insert
137 2.Delete
138 3.Peek
139 4.Display
140 Enter your choice: 1
141
142 Enter the element: 9
143
144 Menu
145 1.Insert
146 2.Delete
147 3.Peek
148 4.Display
149 Enter your choice: 3
150
151 1
152
153 Menu
154 1.Insert
```

```
155 | 2.Delete
156 | 3.Peek
157 | 4.Display
158 | Enter your choice: 4
159 |
160 | 1
161 | 45
162 | 78
163 | 9
164 |
165 | Menu
166 | 1.Insert
167 | 2.Delete
168 | 3.Peek
169 | 4.Display
170 | Enter your choice: 2
171 |
172 | Deleted element: 1
173 |
174 | Menu
175 | 1.Insert
176 | 2.Delete
177 | 3.Peek
178 | 4.Display
179 | Enter your choice: 4
180 |
181 | 45
182 | 78
183 | 9
184 |
185 | Menu
186 | 1.Insert
187 | 2.Delete
188 | 3.Peek
189 | 4.Display
190 | Enter your choice: 3
191 |
192 | 45
193 |
194 | Menu
195 | 1.Insert
196 | 2.Delete
197 | 3.Peek
198 | 4.Display
199 | Enter your choice:
200 | */
```