

Meet patel - DSA - exp 2 – implementation of queue .

```
1
2#include <stdio.h>
3int Q[100], FRONT = -1, REAR = -1, i, n, x, choice;
4void insert();
5void delete ();
6void display();
7
8void main()
9{
10     printf("\t WELCOME to implementation of QUEUE using array !! \n");
11     printf("Enter the size of Queue (Maximum size = 100): ");
12     scanf("%d", &n);
13     do
14     {
15         printf("\n Queue Operation available: \n");
16         printf("\t1.Insert \t2.Delete \t3.Display \t4.Exit \n");
17         printf("\n Enter your choice: ");
18         scanf("%d", &choice);
19         switch (choice)
20         {
21             case 1:
22                 insert();
23                 break;
24             case 2:
25                 delete ();
26                 break;
27             case 3:
28                 display();
29                 break;
30             case 4:
31                 printf("Exit: Program Finished !! ");
32                 break;
33             default:
34                 printf("Please enter a valid choice 1, 2, 3, 4 \n");
35                 break;
36         }
37     } while (choice != 4);
38 }
39
40 void insert()
41 {
42     if (REAR >= n - 1)
43     {
44         printf("Queue Overflow ! \n");
45     }
46     else
47     {
48         printf("Enter the element to insert: ");
49         scanf("%d", &x);
50         REAR++;
51         Q[REAR] = x;
52         if (FRONT == -1)
53         {
54             FRONT = 0;
55         }
56     }
57 }
58
59 void delete ()
60 {
61     if (FRONT == -1)
62     {
63         printf("Queue Underflow ! \n");
64     }
65     else
66     {
67         printf("The deleted element is: %d \n", Q[FRONT]);
68         if (FRONT == REAR)
69             FRONT = REAR = -1;
70         else
71             FRONT++;
72     }
73 }
74
75 void display()
76 {
77     if (REAR < 0)
78     {
79         printf("Queue is empty ! \n");
80     }
81     else
82     {
83         printf("The elements in the Queue are: \n");
84         for (i = FRONT; i < n; i++)
85         {
86             printf("%d ", Q[i]);
87         }
88         printf("\n");
89     }
90 }
```

```
35     }
36     } while (choice != 4);
37 }
38
39 void insert()
40 {
41     if (REAR >= n - 1)
42     {
43         printf("Queue Overflow ! \n");
44     }
45     else
46     {
47         printf("Enter the element to insert: ");
48         scanf("%d", &x);
49         REAR++;
50         Q[REAR] = x;
51         if (FRONT == -1)
52         {
53             FRONT = 0;
54         }
55     }
56 }
57
58 void delete ()
59 {
60     if (FRONT == -1)
61     {
62         printf("Queue Underflow ! \n");
63     }
64     else
65     {
66         printf("The deleted element is: %d \n", Q[FRONT]);
67         if (FRONT == REAR)
68             FRONT = REAR = -1;
69         else
70             FRONT++;
71     }
72 }
73
74 void display()
75 {
76     if (REAR < 0)
77     {
78         printf("Queue is empty ! \n");
79     }
80     else
81     {
82         printf("The elements in the Queue are: \n");
83         for (i = FRONT; i < n; i++)
84         {
85             printf("%d ", Q[i]);
86         }
87         printf("\n");
88     }
89 }
```

```
Open  [?] meet2c Save
31     REAR++;
32     Q[REAR] = x;
33     if (FRONT == -1)
34     {
35         FRONT = 0;
36     }
37 }
38
39 void delete ()
40 {
41     if (FRONT == -1)
42     {
43         printf("Queue Underflow ! \n");
44     }
45     else
46     {
47         printf("The deleted element is: %d \n", Q[FRONT]);
48         if (FRONT == REAR)
49             FRONT = REAR = -1;
50         else
51             FRONT++;
52     }
53 }
54
55 void display()
56 {
57     if (REAR < 0)
58     {
59         printf("Queue is empty ! \n");
60     }
61     else
62     {
63         printf("The elements in the Queue are: \n");
64         for (i = FRONT; i < n; i++)
65         {
66             printf("%d ", Q[i]);
67         }
68         printf("\n");
69     }
70 }
```

```

Enter your choice: 1
Enter the element to insert: 4

Queue Operation available:
1.Insert      2.Delete      3.Display      4.Exit

Enter your choice: 1
Enter the element to insert: 6

Queue Operation available:
1.Insert      2.Delete      3.Display      4.Exit

Enter your choice: 1
Enter the element to insert: 7

Queue Operation available:
1.Insert      2.Delete      3.Display      4.Exit

Enter your choice: 1
Enter the element to insert: 8

Queue Operation available:
1.Insert      2.Delete      3.Display      4.Exit

Enter your choice: 1
Enter the element to insert: 3

Queue Operation available:
1.Insert      2.Delete      3.Display      4.Exit

Enter your choice: 1
Queue Overflow !

Queue Operation available:
1.Insert      2.Delete      3.Display      4.Exit

Enter your choice: 2
The deleted element is: 4

Queue Operation available:
1.Insert      2.Delete      3.Display      4.Exit

Enter your choice: 4
Exit: Program Finished !! d10409@noname:~$

```

```

Queue Operation available:
1.Insert      2.Delete      3.Display      4.Exit

Enter your choice: 4
Exit: Program Finished !! d10409@noname:~$ ./a.out
WELCOME to implementation of QUEUE using array !!
Enter the size of Queue (Maximum size = 100): 5

Queue Operation available:
1.Insert      2.Delete      3.Display      4.Exit

Enter your choice: 1
Enter the element to insert: 3

Queue Operation available:
1.Insert      2.Delete      3.Display      4.Exit

Enter your choice: 1
Enter the element to insert: 5

Queue Operation available:
1.Insert      2.Delete      3.Display      4.Exit

Enter your choice: 1
Enter the element to insert: 7

Queue Operation available:
1.Insert      2.Delete      3.Display      4.Exit

Enter your choice: 2
The deleted element is: 3

Queue Operation available:
1.Insert      2.Delete      3.Display      4.Exit

Enter your choice: 3
The elements in the Queue are:
5 7 0 0

Queue Operation available:
1.Insert      2.Delete      3.Display      4.Exit

Enter your choice: 4
Exit: Program Finished !! d10409@noname:~$

```