

Name-Jeevan R

Section-I

DSA: Lab Program-3

b) WAP to simulate the working of Circular Queue using an array with the following operations: Insert, Delete and Display, also should print appropriate message for queue empty and overflow conditions.

```
#include <stdio.h>

#define SIZE 3

int queue[SIZE];

int front = -1, rear = -1;

void insert() {
    int x;
    if ((rear + 1) % SIZE == front) {
        printf("Queue Overflow!\n");
        return;
    }
    printf("Enter value: ");
    scanf("%d", &x);
    if (front == -1) front = rear = 0;
    else rear = (rear + 1) % SIZE;
    queue[rear] = x;
    printf("%d inserted.\n", x);
}

void delete() {
    if (front == -1) {
        printf("Queue Underflow!\n");
        return;
    }
}
```

```
    }  
    printf("%d deleted.\n", queue[front]);  
    if (front == rear) front = rear = -1;  
    else front = (front + 1) % SIZE;  
}
```

```
void display() {  
    if (front == -1) {  
        printf("Queue is empty.\n");  
        return;  
    }  
}
```

```
printf("Queue: ");  
int i = front;  
while (1) {  
    printf("%d ", queue[i]);  
    if (i == rear) break;  
    i = (i + 1) % SIZE;  
}  
printf("\n");  
}
```

```
int main() {  
    int choice;  
    printf("---Circular Queue---");  
    while (1) {  
        printf("\n1.Insert 2.Delete 3.Display 4.Exit\n Enter Choice: ");  
        scanf("%d", &choice);  
        switch(choice) {  
            case 1: insert(); break;  
            case 2: delete(); break;
```

case 3: display(); break;

case 4: return 0;

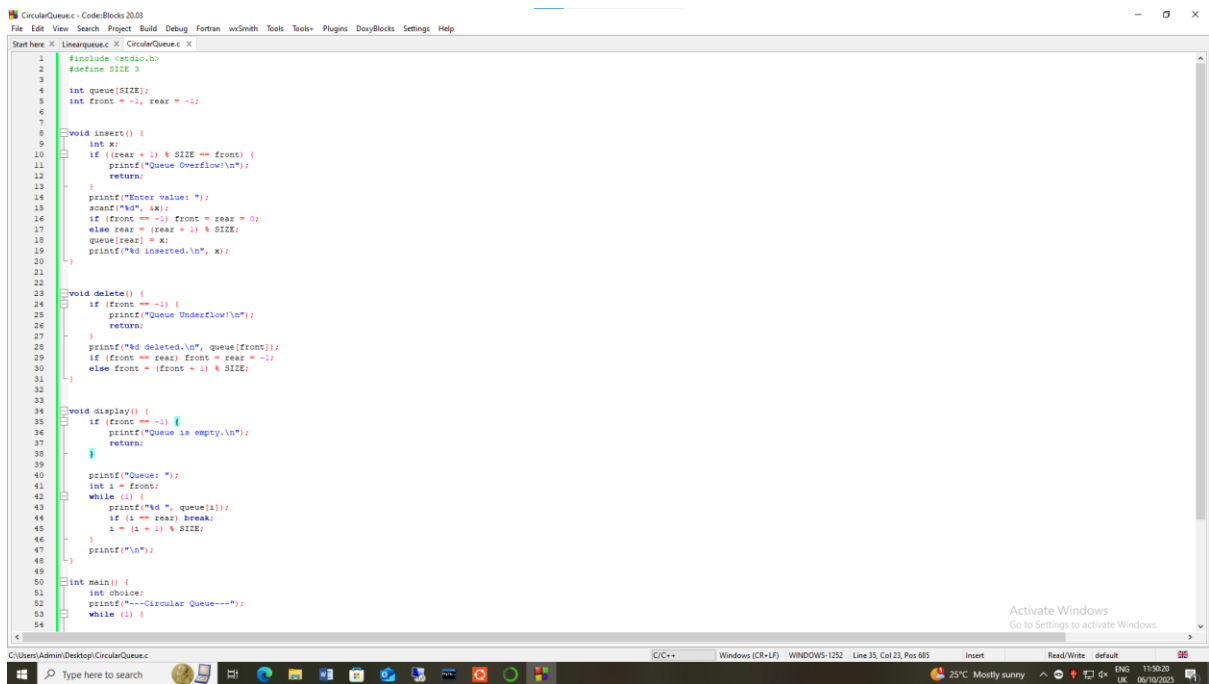
default: printf("Invalid choice!\n");

}

}

}

Code and Expected Output:



```
1 #include <stdio.h>
2 #define SIZE 5
3
4 int queue[SIZE];
5 int front = -1, rear = -1;
6
7
8 void insert() {
9     int x;
10    if ((rear == 1) & SIZE == front) {
11        printf("Queue Overflow!\n");
12        return;
13    }
14    printf("Enter value: ");
15    scanf("%d", &x);
16    if (front == -1) front = rear = 0;
17    else rear = (rear + 1) % SIZE;
18    queue[rear] = x;
19    printf("%d inserted.\n", x);
20 }
21
22 void delete() {
23     if (front == -1) {
24         printf("Queue Underflow!\n");
25         return;
26     }
27     printf("%d deleted.\n", queue[front]);
28     if (front == rear) front = rear = -1;
29     else front = (front + 1) % SIZE;
30 }
31
32
33 void display() {
34     if (front == -1) {
35         printf("Queue is empty.\n");
36         return;
37     }
38     printf("Queue: ");
39     int i = front;
40     while (1) {
41         printf("%d ", queue[i]);
42         if (i == rear) break;
43         i = (i + 1) % SIZE;
44     }
45     printf("\n");
46 }
47
48
49 int main() {
50     int choice;
51     printf("----Circular Queue----\n");
52     while (1) {
```

```

#include <stdio.h>
#define SIZE 3

int queue[SIZE];
int front = -1, rear = -1;

void insert() {
    int x;
    if ((rear + 1) % SIZE == front) {
        printf("Queue Overflow!\n");
        return;
    }
    printf("Enter value: ");
    scanf("%d", &x);
    if (front == -1) front = rear = 0;
    else rear = (rear + 1) % SIZE;
    queue[rear] = x;
    printf("%d inserted.\n", x);
}

void delete() {
    if (front == -1) {
        printf("Queue Underflow!\n");
        return;
    }
    printf("%d deleted.\n", queue[front]);
    if (front == rear) front = rear = -1;
    else front = (front + 1) % SIZE;
}

void display() {
    if (front == -1) {
        printf("Queue is empty.\n");
        return;
    }

    printf("Queue: ");
    int i = front;
    while (1) {
        printf("%d ", queue[i]);
        if (i == rear) break;
        i = (i + 1) % SIZE;
    }
    printf("\n");
}

int main() {
    int choice;
    printf("---Circular Queue---");
    while (1) {

```

```

46         }
47         printf("\n");
48     }
49
50     int main() {
51         int choice;
52         printf("---Circular Queue---");
53         while (1) {
54
55             printf("\n1.Insert 2.Delete 3.Display 4.Exit\n Enter Choice: ");
56             scanf("%d", &choice);
57             switch(choice) {
58                 case 1: insert(); break;
59                 case 2: delete(); break;
60                 case 3: display(); break;
61                 case 4: return 0;
62                 default: printf("Invalid choice!\n");
63             }
64         }
65     }
66

```

```
C:\Users\Admin\Desktop\CircularQueue.exe
---Circular Queue---
1.Insert 2.Delete 3.Display 4.Exit
Enter Choice: 3
Queue is empty.

1.Insert 2.Delete 3.Display 4.Exit
Enter Choice: 1
Enter value: 10
10 inserted.

1.Insert 2.Delete 3.Display 4.Exit
Enter Choice: 1
Enter value: 20
20 inserted.

1.Insert 2.Delete 3.Display 4.Exit
Enter Choice: 1
Enter value: 30
30 inserted.

1.Insert 2.Delete 3.Display 4.Exit
Enter Choice: 1
Queue Overflow!

1.Insert 2.Delete 3.Display 4.Exit
Enter Choice: 3
Queue: 10 20 30

1.Insert 2.Delete 3.Display 4.Exit
Enter Choice: 2
10 deleted.

1.Insert 2.Delete 3.Display 4.Exit
Enter Choice: 2
20 deleted.

1.Insert 2.Delete 3.Display 4.Exit
Enter Choice: 3
Queue: 30

1.Insert 2.Delete 3.Display 4.Exit
Enter Choice: 1
Enter value: 40
40 inserted.

1.Insert 2.Delete 3.Display 4.Exit
Enter Choice: 3
Queue: 30 40

1.Insert 2.Delete 3.Display 4.Exit
Enter Choice: 2
30 deleted.

1.Insert 2.Delete 3.Display 4.Exit
Enter Choice: 2
40 deleted.

1.Insert 2.Delete 3.Display 4.Exit
Enter Choice: 2
Queue Underflow!

1.Insert 2.Delete 3.Display 4.Exit
Enter Choice: 3
```

```
1.Insert 2.Delete 3.Display 4.Exit
Enter Choice: 3
Queue: 30 40

1.Insert 2.Delete 3.Display 4.Exit
Enter Choice: 2
30 deleted.

1.Insert 2.Delete 3.Display 4.Exit
Enter Choice: 2
40 deleted.

1.Insert 2.Delete 3.Display 4.Exit
Enter Choice: 2
Queue Underflow!

1.Insert 2.Delete 3.Display 4.Exit
Enter Choice: 3
Queue is empty.

1.Insert 2.Delete 3.Display 4.Exit
Enter Choice: 4

Process returned 0 (0x0)   execution time : 80.723 s
Press any key to continue.
```