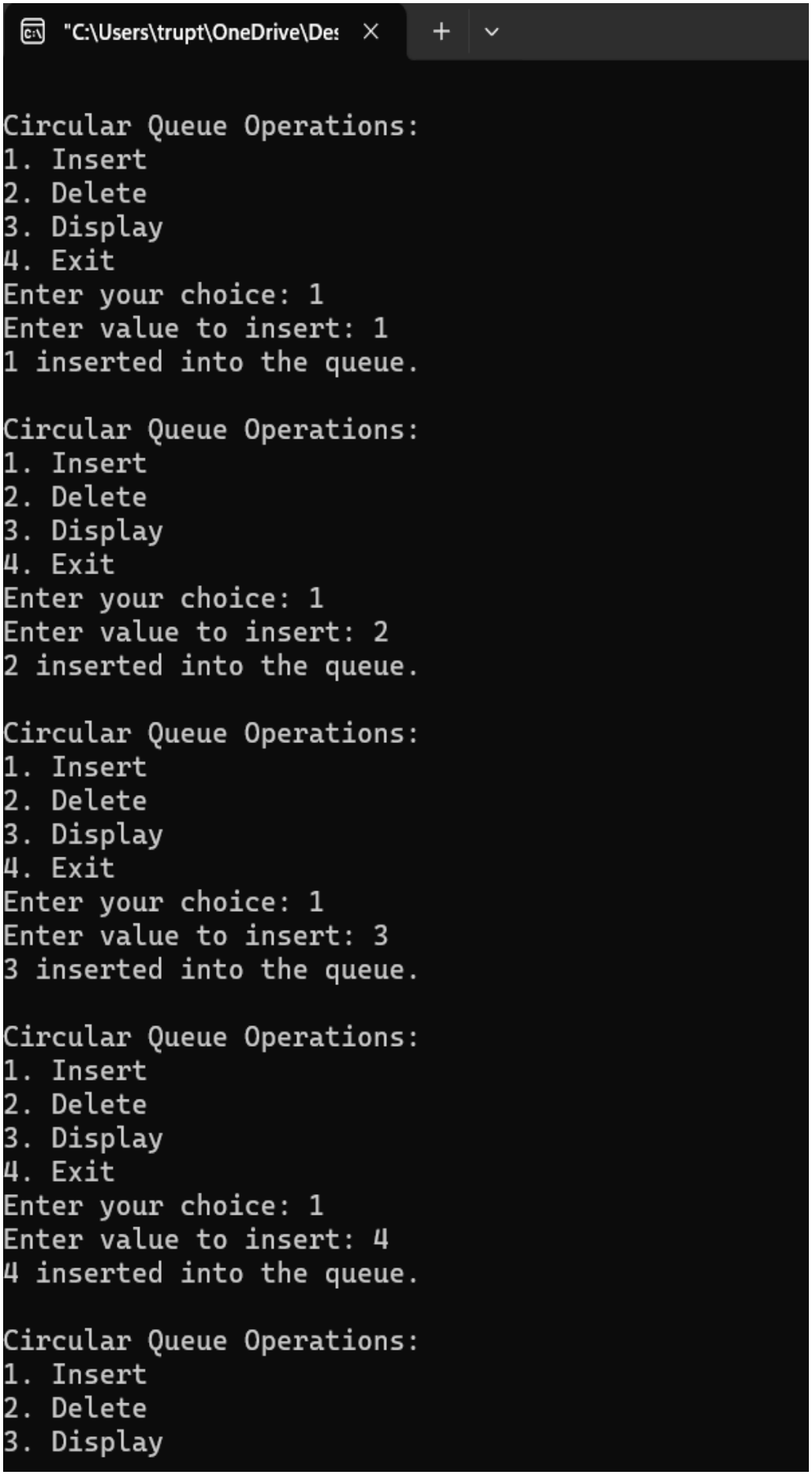
DS LAB PROGRAM 3B

Start here | Labprogram3B.c

```c
#include <stdio.h>
#define MAX 5 // maximum size of the queue
int queue[MAX];
int front = -1, rear = -1;
// Function to insert an element in the circular queue
void insert(int value)
{
if ((front == 0 && rear == MAX - 1) || (front == (rear + 1) % MAX))
{
printf("Queue Overflow! Cannot insert %d\n", value);
}
else
{
if (front == -1)
{ // first insertion
front = 0;
rear = 0;
}
else
{
rear = (rear + 1) % MAX;
}
queue[rear] = value;
printf("%d inserted into the queue.\n", value);
}
}
// Function to delete an element from the circular queue
void delete()
{
if (front == -1)
{
printf("Queue Underflow! Queue is empty.\n");
}
else
{
printf("Deleted element: %d\n", queue[front]);
```

```
36  printf("Deleted element: %d\n", queue[front]);
37  if (front == rear)
38  {
39  // queue becomes empty
40  front = -1;
41  rear = -1;}
42  else
43  {
44  front = (front + 1) % MAX;
45  }
46  }
47  }
48  // Function to display elements of the circular queue
49  void display()
50  {
51  if (front == -1)
52  {
53  printf("Queue is empty.\n");
54  }
55  else
56  {
57  printf("Queue elements: ");
58  int i = front;
59  while (1)
60  {
61  printf("%d ", queue[i]);
62  if (i == rear)
63  break;
64  i = (i + 1) % MAX;
65  }
66  printf("\n");
67  }
68  }
69  int main()
70  {
71  int choice, value;
```

```c
}
int main()
{
int choice, value;
while (1)
{
printf("\nCircular Queue Operations:\n");
printf("1. Insert\n");
printf("2. Delete\n");
printf("3. Display\n");
printf("4. Exit\n");
printf("Enter your choice: ");
scanf("%d", &choice);
switch (choice)
{
case 1:
printf("Enter value to insert: ");
scanf("%d", &value);
insert(value);break;
case 2:
delete();
break;
case 3:
display();
break;
case 4:
printf("Exiting program.\n");
return 0;
default:
printf("Invalid choice! Please try again.\n");
}
}
return 0;
}
```


"C:\Users\trupt\OneDrive\De
Circular Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter value to insert: 1
1 inserted into the queue.
Circular Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter value to insert: 2
2 inserted into the queue.
Circular Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter value to insert: 3
3 inserted into the queue.
Circular Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter value to insert: 4
4 inserted into the queue.
Circular Queue Operations:
1. Insert
2. Delete
3. Display

```
4 inserted into the queue.

Circular Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter value to insert: 5
5 inserted into the queue.

Circular Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 3
Queue elements: 1 2 3 4 5

Circular Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter value to insert: 1
Queue Overflow! Cannot insert 1

Circular Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 2
Deleted element: 1

Circular Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
```

```
Enter your choice: 2
Deleted element: 3

Circular Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 2
Deleted element: 4

Circular Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 2
Deleted element: 5

Circular Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 2
Queue Underflow! Queue is empty.

Circular Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 3
Queue is empty.

Circular Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
```

```
Circular Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter value to insert: 6
6 inserted into the queue.

Circular Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter value to insert: 7
7 inserted into the queue.

Circular Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter value to insert: 8
8 inserted into the queue.

Circular Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter value to insert: 9
9 inserted into the queue.

Circular Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
```

```
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter value to insert: 10
10 inserted into the queue.

Circular Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter value to insert: 2
Queue Overflow! Cannot insert 2

Circular Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 3
Queue elements: 6 7 8 9 10

Circular Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 4
Exiting program.

Process returned 0 (0x0)   execution time : 174.947 s
Press any key to continue.
```