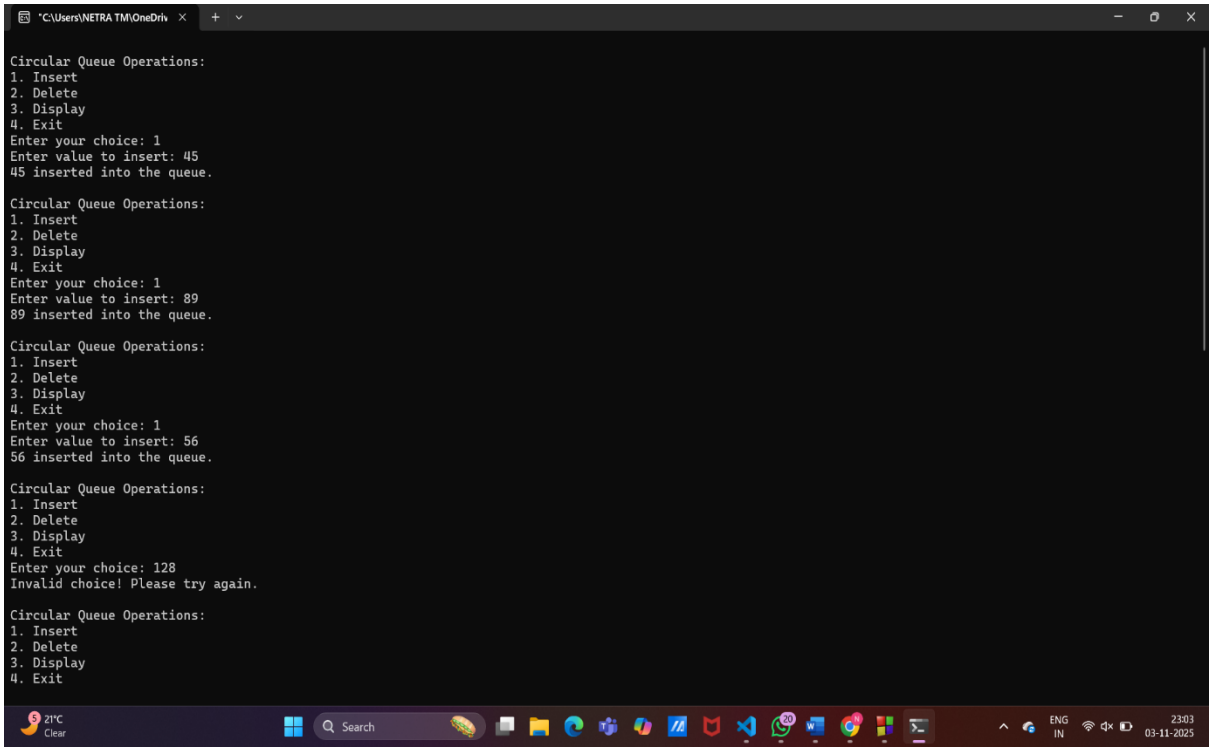
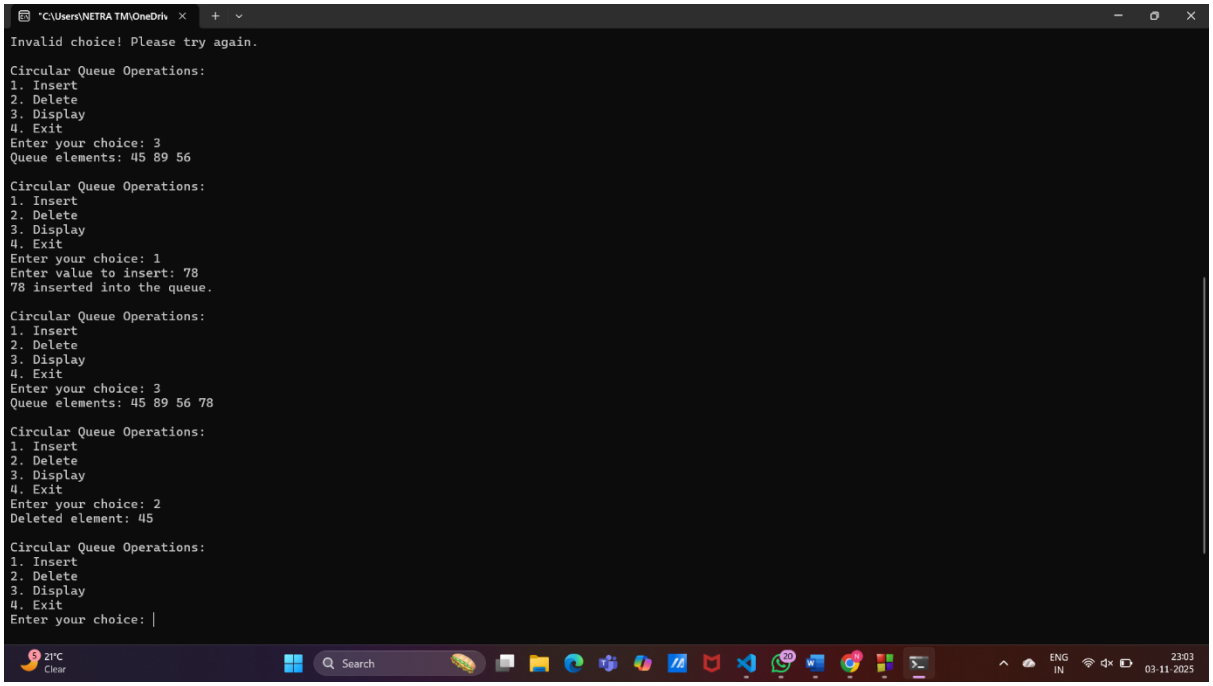
```
#include <stdio.h>
#define MAX 5 // maximum size of the queue
int queue[MAX];
int front = -1, rear = -1;
// Function to insert an element in the circular queue
void insert(int value)
{
 if ((front == 0 && rear == MAX - 1) || (front == (rear + 1) % MAX))
 {
 printf("Queue Overflow! Cannot insert %d\n",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

```
 if (i == rear)
 break;
 i = (i + 1) % MAX;
 }
 printf("\n");
 }
}
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\NETRA TM\OneDriv
Invalid choice! Please try again.
Circular Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 3
Queue elements: 45 89 56
Circular Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter value to insert: 78
78 inserted into the queue.
Circular Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 3
Queue elements: 45 89 56 78
Circular Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 2
Deleted element: 45
Circular Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice:
21°C
Clear
Search
ENG
IN
23:03
03-11-2025