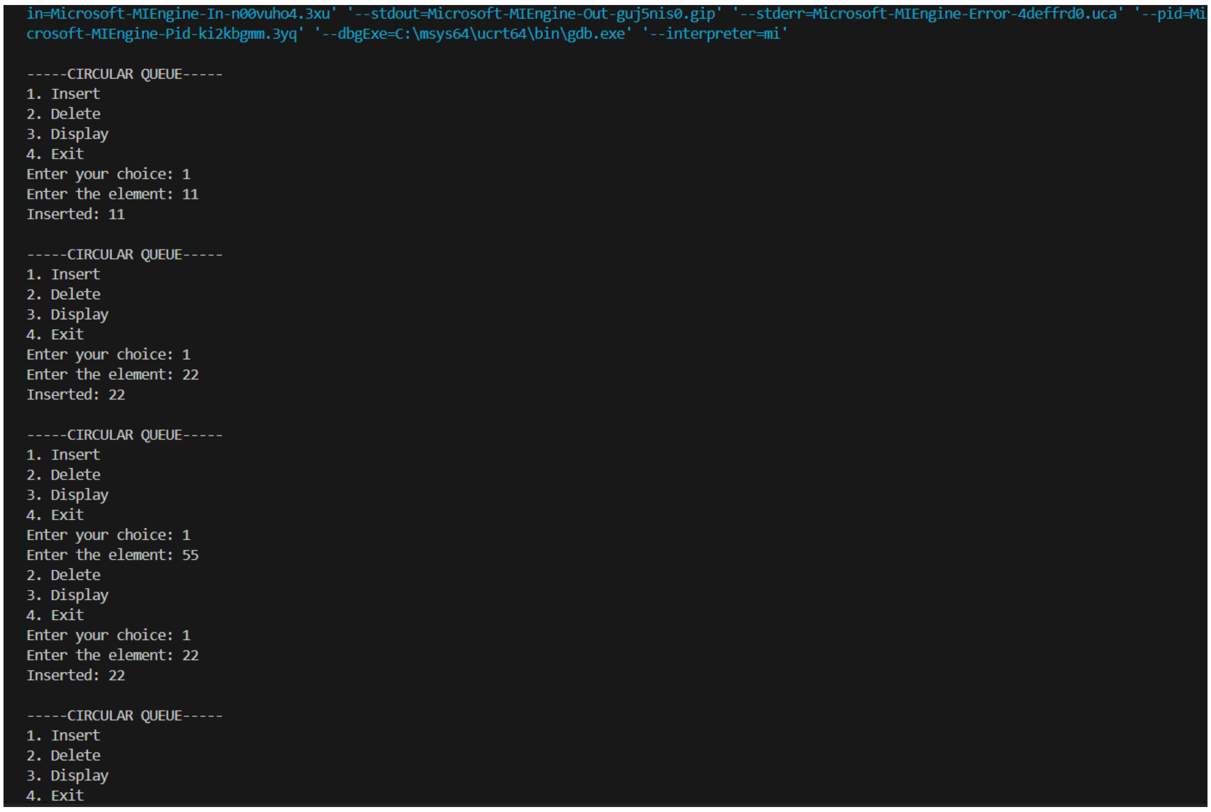
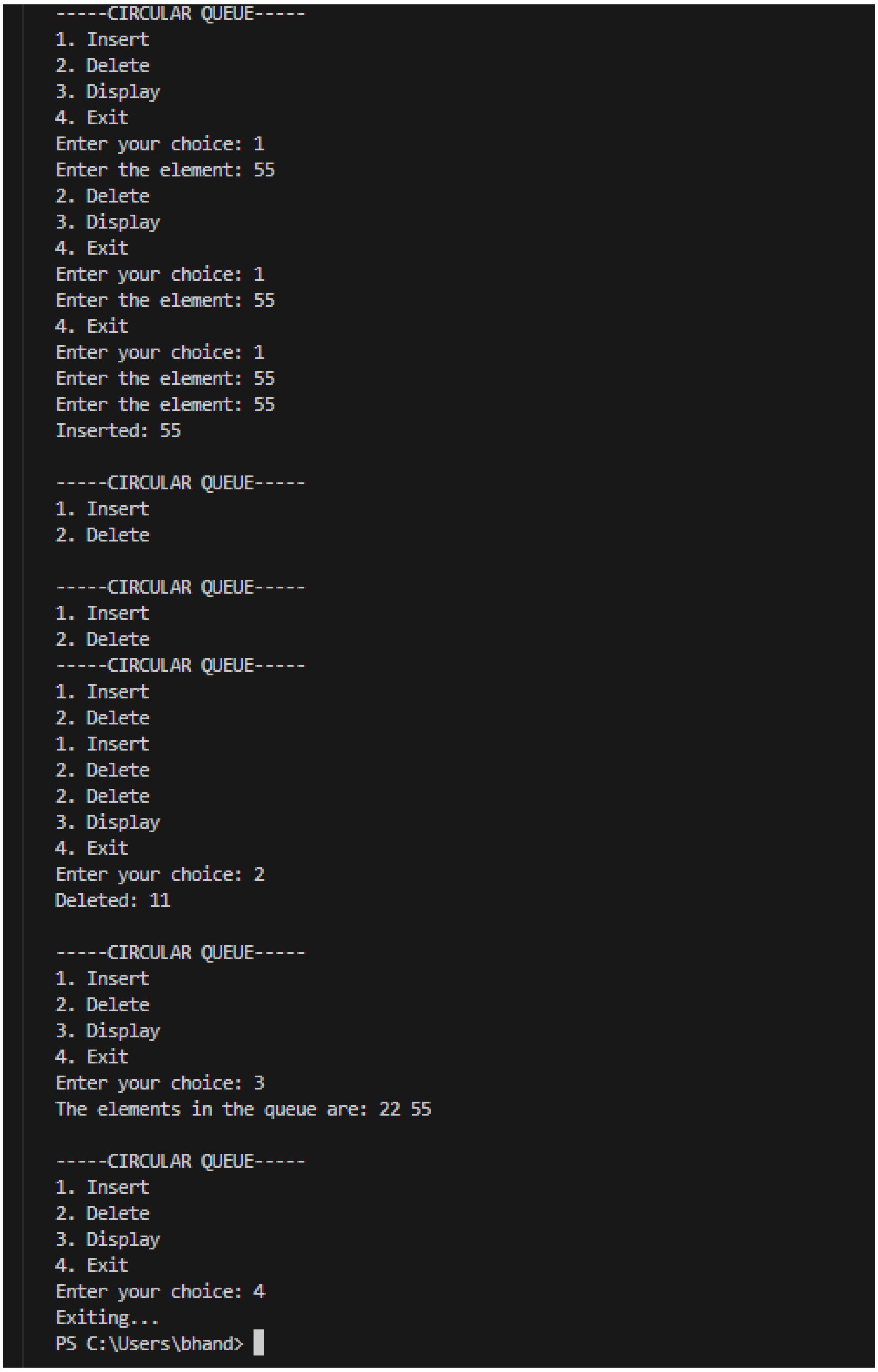
```
#include <stdio.h>
#define MAXSIZE 10

int cq[MAXSIZE];
int rear = -1, front = -1;

void insert();
void delete1();
void display();

void main() {
   int choice;
   do {
      printf("\n-----CIRCULAR QUEUE-----");
      printf("\n1. Insert \n2. Delete \n3. Display \n4. Exit");
      printf("\nEnter your choice: ");
      scanf("%d", &choice);

      switch (choice) {
         case 1:
            insert();
            break;
         case 2:
            delete1();
            break;
         case 3:
            display();
            break;
         case 4:
            printf("Exiting...\n");
            break;
            default:
                printf("Invalid choice. Try again.\n");
        }
    } while (choice != 4);
}

void insert() {
    int n;
    if ((front == (rear + 1) % MAXSIZE)) { // Condition for queue full
        printf("Queue is overflow\n");
    } else {
        printf("Enter the element: ");
        scanf("%d", &n);
        if (rear == -1 && front == -1) { // Initial insertion
            rear = front = 0;
        } else {
            rear = (rear + 1) % MAXSIZE; // Increment rear circularly
        }
        cq[rear] = n;
        printf("Inserted: %d\n", n);
    }
}

void delete1() {
    if (rear == -1 && front == -1) { // Condition for queue empty
        printf("Queue is empty\n");
    } else {
        int n = cq[front];
        if (front == rear) { // Single element case
            rear = front = -1;
        } else {
            front = (front + 1) % MAXSIZE; // Increment front circularly
        }
        printf("Deleted: %d\n", n);
    }
}

void display() {
    if (rear == -1 && front == -1) { // Queue empty
        printf("Queue is empty\n");
    } else {
        printf("The elements in the queue are: ");
        int i = front;
        while (1) {
            printf("%d ", cq[i]);
            if (i == rear) // Stop when the end is reached
                break;
            i = (i + 1) % MAXSIZE; // Increment circularly
        }
        printf("\n");
    }
```

```
}
```



```
in=Microsoft-MIEngine-In-n00vuho4.3xu' '--stdout=Microsoft-MIEngine-Out-guj5nis0.gip' '--stderr=Microsoft-MIEngine-Error-4deffrd0.uca' '--pid=Mi
crosoft-MIEngine-Pid-ki2kbgmm.3yq' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'

-----CIRCULAR QUEUE-----
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter the element: 11
Inserted: 11

-----CIRCULAR QUEUE-----
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter the element: 22
Inserted: 22

-----CIRCULAR QUEUE-----
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter the element: 55
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter the element: 22
Inserted: 22

-----CIRCULAR QUEUE-----
1. Insert
2. Delete
3. Display
4. Exit
```


```
-----CIRCULAR QUEUE-----
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter the element: 55
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter the element: 55
4. Exit
Enter your choice: 1
Enter the element: 55
Enter the element: 55
Inserted: 55

-----CIRCULAR QUEUE-----
1. Insert
2. Delete

-----CIRCULAR QUEUE-----
1. Insert
2. Delete
-----CIRCULAR QUEUE-----
1. Insert
2. Delete
1. Insert
2. Delete
2. Delete
3. Display
4. Exit
Enter your choice: 2
Deleted: 11

-----CIRCULAR QUEUE-----
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 3
The elements in the queue are: 22 55

-----CIRCULAR QUEUE-----
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 4
Exiting...
PS C:\Users\bhand> 
```