DEMONSTRATE THE WORKING OF A CIRCULAR QUEUE: -

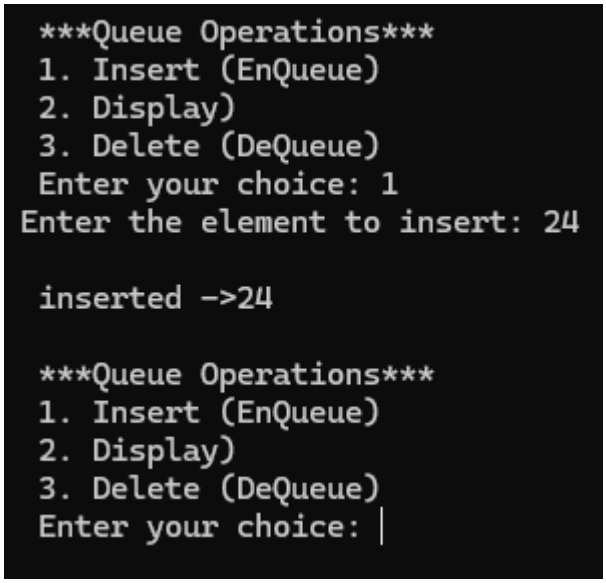
```c
#include <stdio.h>
#define SIZE 5

int items[SIZE];
int front=-1;
int rear=-1;

int isFull() {
if((front==rear+1)||(front==0&&rear==SIZE-1))
    return 1;
    return 0;
}
int isEmpty() {
if (front==-1)
    return 1;
    return 0;
}
void enQueue(int element) {
if(isFull())
    printf("\n Queue is Full!! \n");
    else {
        if(front==-1)
            front=0;
        rear=(rear+1)%SIZE;
        items[rear]=element;
        printf("\n inserted ->%d",element);
    }
}
int deQueue() {
int element;
if (isEmpty()) {
    printf("\n Queue is empty!!!\n");
    return(-1);
} else {
element=items[front];
if(front==rear) {
    front=-1;
```

```
38          rear=-1;
39      }
40      else {
41          front=(front ++)%SIZE;
42      }
43      printf("\n Deleted element ->%d \n", element);
44      return(element);
45      }
46      }
47      void display() {
48      int i;
49      if(isEmpty())
50          printf("\n Empty Queue \n");
51          else {
52              printf("\n Front ->%d",front);
53              printf("\n Items ->");
54              for(i=front;i!=rear;i=(i+1)%SIZE) {
55                  printf("%d", items[i]);
56                  printf("%d",rear);
57              }
58          }
59      }
60      int main() {
61      int choice;
62      int yum;
63      while(1) {
64          printf("\n\n ***Queue Operations***");
65          printf("\n 1. Insert (EnQueue)");
66          printf("\n 2. Display)");
67          printf("\n 3. Delete (DeQueue)");
68          printf("\n Enter your choice: ");
69          scanf("%d",&choice);
70          switch(choice) {
71      case 1:
72          printf("Enter the element to insert: ");
73          scanf("%d",&yum);
74          enQueue(yum);
75          break;
76      case 2:
77          display();
78          break;
79      case 3:
80          deQueue();
81          break;
82      default:
83          printf("Invalid choice. Please try again!! \n");
84      }
85      }
86      }
```

OUTPUT: -


Queue Operations
1. Insert (EnQueue)
2. Display)
3. Delete (DeQueue)
Enter your choice: 1
Enter the element to insert: 24
inserted ->24
Queue Operations
1. Insert (EnQueue)
2. Display)
3. Delete (DeQueue)
Enter your choice: