**Assignment No : 3**

Name: Rasika Mahure

PRN:72018258B

Div:B

Roll No: S1951093

**Aim:-**

Implement Circular Queue using Array. Perform following operations on it. a) Insertion (Enqueue)

1. Deletion (Dequeue)
2. Display

(Note: Handle queue full condition by considering a fixed size of a queue.)

**//PROGRAM**

#include <iostream> using namespace std; int cqueue[5]; int front = -1, rear = -1, n =

5; void insertCQ(int val)

{ if ((front == 0 && rear == n - 1) || (front == rear + 1))

{ cout << "Queue Overflow \n"; return; } if (front == -1) { front = 0; rear = 0;

} else { if (rear == n - 1) rear = 0; else rear = rear + 1;

} cqueue[rear]

= val; }

void deleteCQ()

{ if (front == -

1) {

cout << "Queue Underflow\n";

return; }

cout << "Element deleted from queue is : " << cqueue[front] << endl;

if (front == rear)

{ front

= -1; rear

= -1;

} else { if (front == n - 1) front = 0; else front = front + 1; } }

void displayCQ\_forward() { int f = front, r = rear; if

(front == -1) { cout <<

"Queue is empty" << endl; return; } cout << "Queue elements are :\n"; if (f <= r) { while (f <= r) { cout << cqueue[f] << " "; f++; }

} else

{ while (f <= n - 1) { cout << cqueue[f] << " "; f++; } f = 0; while (f <= r) { cout << cqueue[f] << " "; f++; } } cout << endl; } void displayCQ\_reverse() { int f = front, r =

rear; if (front == -1)

{ cout << "Queue is empty" << endl; return; } cout << "Queue elements are:\n";

if (f <= r)

{ while (f <= r) { cout << cqueue[r] << " "; r--; }

} else

{ while (r >= 0) { cout << cqueue[r] << " "; r--; } r = n - 1; while (r >= f) { cout << cqueue[r] << " "; r--; } } cout << endl;

} int main() { int ch, val; cout << "1)Insert\n"; cout <<

"2)Delete\n"; cout <<

"3)DisplayForward\n"; cout <<

"4)DisplayReverse\n"; cout << "5)Exit\n"; do { cout << "Enter choice : " << endl; cin >> ch; switch (ch) { case 1:

cout << "Input for insertion: " << endl; cin >> val; insertCQ(val); break; case 2: deleteCQ(); break; case 3:

displayCQ\_forward(); break; case 4: displayCQ\_reverse(); break; case 5: cout << "Exit\n"; break; default:

cout << "Incorrect!\n"; } } while (ch != 5); return

0;

}

**//OUTPUT**

1)Insert

2)Delete

3)DisplayForward

4)DisplayReverse 5)Exit Enter choice :

1

Input for insertion:

2 Enter choice :

1

Input for insertion:

4 Enter choice :

1

Input for insertion:

6 Enter choice :

1

Input for insertion:

8 Enter choice :

1

Input for insertion:

9

Enter choice :

3

Queue elements are : 2 4 6 8 9 Enter choice :

2

Element deleted from queue is : 2 Enter choice :

4

Queue elements are:

9 8 6 4

Enter choice :

5

Exit