**DSA Assignment: 4**

**Exp 4:** [Implementation of Circular Queue Data Structure using array.](https://classroom.google.com/c/NDg4NTg4NTM0NDIz/a/NTM4MDQwNzg0MzIx/details" \t "_self)

Shashwat Tripathi

D10A Roll No: 60

**AIM:** In this experiment, with the help of array we will implement Circular Queue data structure

**CODE:**

// Exp 04 Implementation of Circular Queue Data Structure using array.

#include <stdio.h>

#include <stdlib.h>

#define max 6

int queue[max];

int front = -1;

int rear = -1;

void enqueue(int element);

int dequeue();

void display();

int main()

{

    printf("D10A\_60\_Shashwat Tripathi\n");

    int choice = 1, x;

    printf("\n#############################################\n");

    printf("\nPress 1: Insert an element");

    printf("\nPress 2: Delete an element");

    printf("\nPress 3: Display the element");

    printf("\nPress 4: Exit");

    printf("\n#############################################\n");

    while (choice < 5 && choice != 0)

    {

        printf("\nEnter your choice:");

        scanf("%d", &choice);

        switch (choice)

        {

        case 1:

            printf("Enter the element :");

            scanf("%d", &x);

            enqueue(x);

            break;

        case 2:

            dequeue();

            break;

        case 3:

            display();

            break;

        case 4:

            exit(0);

            break;

        default:

            printf("Invalid Choice");

        }

    }

    return 0;

}

void enqueue(int element)

{

    if (front == -1 && rear == -1)

    {

        front = 0;

        rear = 0;

        queue[rear] = element;

    }

    else if ((rear + 1) % max == front)

    {

        printf("Queue is overflow..");

    }

    else

    {

        rear = (rear + 1) % max;

        queue[rear] = element;

    }

}

int dequeue()

{

    if ((front == -1) && (rear == -1))

    {

        printf("\nQueue is underflow..");

    }

    else if (front == rear)

    {

        printf("\nThe dequeued element is %d", queue[front]);

        front = -1;

        rear = -1;

    }

    else

    {

        printf("\nThe dequeued element is %d", queue[front]);

        front = (front + 1) % max;

    }

}

void display()

{

    int i = front;

    if (front == -1 && rear == -1)

    {

        printf("\n Queue is empty..");

    }

    else

    {

        printf("\nElements in a Queue are :");

        while (i <= rear)

        {

            printf("%d,", queue[i]);

            i = (i + 1) % max;

        }

    }

}

**OUTPUT:**

