**DSA Assignment: 3**

**Exp 3:** [Implementation of Linear Queue Data Structure using array.](https://classroom.google.com/c/NDg4NTg4NTM0NDIz/a/NTM4MDQyOTQ1NzU4/details" \t "_self)

Shashwat Tripathi

D10A Roll No: 60

**AIM:** In this experiment, with the help of the concept of arrays we will implement Linear Queue Data Structure.

**CODE:**

// Exp 03 Implementation of Linear Queue Data Structure using array.

#include <stdio.h>

#include <stdlib.h>

#define MAX 10

int queue[MAX];

int front = -1, rear = -1;

void insert();

void delete ();

void display();

int main()

{

    int choice;

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

    printf("################################################");

    printf("\n Choices are:\n 1.Insert\n 2.Delete\n 3.Display\n 4.Exit\n");

    printf("################################################");

    do

    {

        printf("\nEnter your choice: ");

        scanf("%d", &choice);

        switch (choice)

        {

        case 1:

            insert();

            break;

        case 2:

            delete ();

            break;

        case 3:

            printf("\nElements in the Queue are: \n");

            display();

            break;

        case 4:

            break;

        }

    } while (choice != 4);

    return 0;

}

void insert()

{

    int n;

    printf("Enter the element to be inserted in the queue: ");

    scanf("%d", &n);

    if (rear == MAX - 1)

        printf("Overflow Condition\n");

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

    {

        front = rear = 0;

    }

    else

    {

        rear = rear + 1;

    }

    queue[rear] = n;

}

void delete ()

{

    int Val;

    if (front == -1 || front > rear)

    {

        printf("Underflow Condition\n");

    }

    else

    {

        Val = queue[front];

        if (front == rear)

        {

            front = rear = -1;

        }

        else

        {

            front++;

        }

        printf("Value is Deleted \n");

    }

}

void display()

{

    int i;

    printf("\n");

    if (front == -1 || front > rear)

        printf("Queue is Empty\n");

    else

    {

        for (i = front; i <= rear; i++)

        {

            printf("%d \n", queue[i]);

        }

    }

}

**OUPUT:**

