**Experiment – 3 – Code and Output**

**Question – 1 – Linear Queue: -**

**Code:**

#include <stdio.h>

#include <stdlib.h>

#define MAX 5

void Enqueue(int);

void Dequeue();

void QueueFront();

void QueueRear();

void Display();

int f = -1, r = -1, queue[MAX];

int main()

{

    int choice, data;

    do

    {

        printf("\nEnter choice\n1. Enqueue\n2. Dequeue\n3. Front of Queue\n4. Rear of Queue\n5. Display\n6. Exit\t");

        printf("\nEnter choice : ");

        scanf("%d", &choice);

        switch (choice)

        {

            case 1:

                printf("\nEnter data: ");

                scanf("%d", &data);

                Enqueue(data);

                Display();

                break;

            case 2:

                Dequeue();

                Display();

                break;

            case 3:

                QueueFront();

                break;

            case 4:

                QueueRear();

                break;

            case 5:

                Display();

                break;

            case 6:

                exit(0);

                break;

            default:

                printf("\nEnter a valid choice!");

        }

    } while (choice != 6);

    return 0;

}

void Enqueue(int info)

{

    if(r == (MAX - 1))

        printf("\nQueue is Full");

    else if(f == -1) {

        f = r = 0;

        queue[r] = info;

    }

    else

        queue[++r] = info;

}

void Dequeue()

{

    if (f == -1)

        printf("\nQueue is Empty");

    else {

        printf("\nElement Dequeued: %d\n", queue[f]);

        if (f == r)

            f = r = -1;

        else

            f++;

    }

}

void QueueFront()

{

    if (f == -1)

        printf("\nQueue is Empty");

    else

        printf("\nFront of queue: %d", queue[f]);

}

void QueueRear()

{

    if (r == -1)

        printf("\nQueue is Empty");

    else

        printf("\nRear of queue: %d", queue[r]);

}

void Display()

{

    int i;

    if(f == -1)

        printf("\nQueue is Empty");

    else {

        for(i = f ; i <= r ; i++) {

            printf("%d", queue[i]);

            if(i < r)

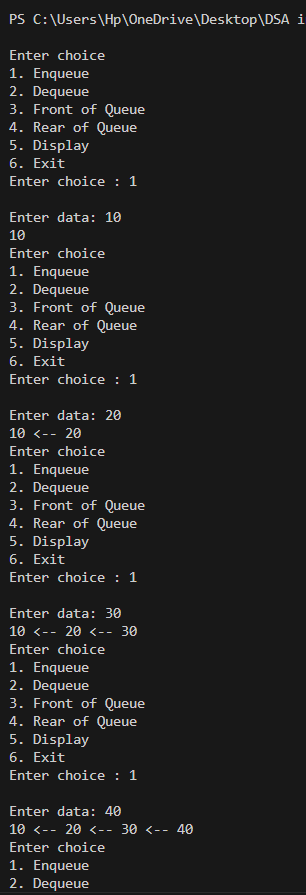
                printf(" <-- ");

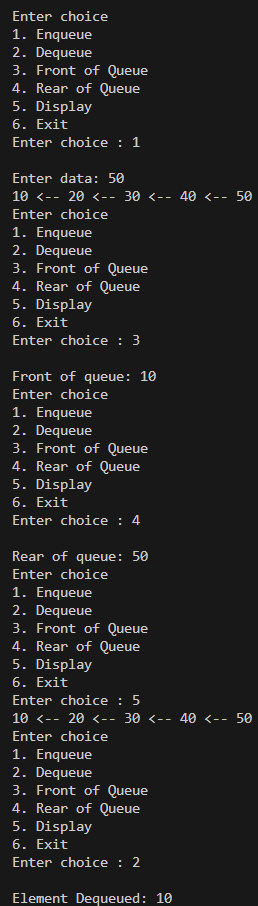
        }

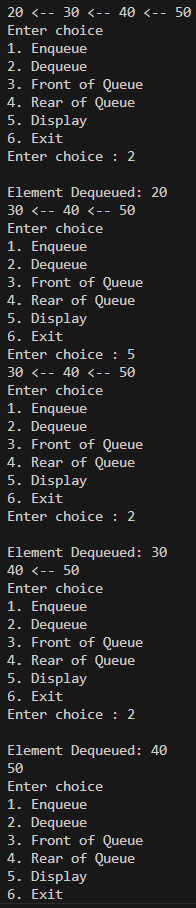
    }

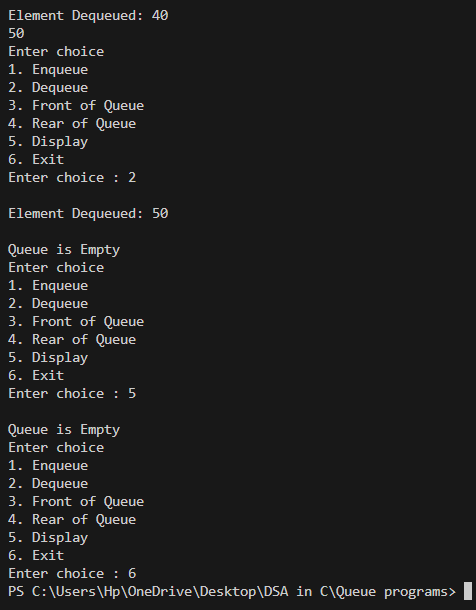
}

**Output:**









**Question – 2 – Circular Queue: -**

**Code:**

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#define MAX 5

void Enqueue(int);

void Dequeue();

void Display();

int f = -1, r = -1, queue[MAX];

int main()

{

    int choice, data;

    do

    {

        printf("\nEnter choice\n1. Enqueue\n2. Dequeue\n3. Display\n4. Exit\t");

        printf("\nEnter your choice = ");

        scanf("%d", &choice);

        switch(choice)

        {

            case 1: printf("\nEnter data : ");

                    scanf("%d", &data);

                    Enqueue(data);

                    Display();

                    break;

            case 2: Dequeue();

                    Display();

                    break;

            case 3: Display();

                    break;

            case 4: exit(0);

                    break;

            default: printf("\nEnter a valid choice");

        }

    }

    while(choice != 4);

    getch();

    return 0;

}

void Enqueue(int info)

{

    if(f == -1 && r == -1) {

        f = r = 0;

        queue[r] = info;

    }

    else if((r+1)%MAX == f)

        printf("\nQueue is Full");

    else {

        r = (r+1)%MAX;

        queue[r] = info;

    }

}

void Dequeue()

{

    if(f == -1 && r == -1)

        printf("\nQueue is Empty");

    else if(f == r) {

        printf("\nElement Dequeued : %d", queue[f]);

        f = r = -1;

    }

    else {

        printf("\nElement Dequeued : %d", queue[f]);

        f = (f+1)%MAX;

    }

}

void Display()

{

    int i;

    printf("\n");

    if(f == -1 && r == -1)

        printf("\nQueue is Empty!");

    else {

        if(f > r) {

            for(i=f ; i<MAX ; i++)

                printf("%d <-- ", queue[i]);

            for(i=0 ; i<=r ; i++)

                printf("%d <-- ", queue[i]);

        }

        else {

            for(i=f ; i<=r ; i++)

                printf("%d <-- ", queue[i]);

        }

    }

}

**Output:**

