

DAY 5 – DEQUEUE

7. Write a menu driven C program to implement DEQUEUE using arrays and perform the following operations

- a. Insert from the front
- b. Insert from rear
- c. Delete from front
- d. Delete from rear
- e. Display.

PROGRAM

```
#include<stdio.h>
#define MAX 10
int dequeue[MAX], front = -1, rear = -1, i;
void insert_rear()
{
    int n;
    if(front == rear + 1 || (front == 0 && rear == MAX - 1))
    {
        printf("\nQueue is full.\n"); return;
    }
    printf("Enter element to insert: ");
    scanf("%d", &n);
    if(front == -1)
        front = 0;
    rear = (rear + 1) % MAX;
    dequeue[rear] = n;
}
void insert_front()
{
    int n;
    if(front == rear + 1 || (front == 0 && rear == MAX - 1))
    {
        printf("\nQueue is full.\n");
        return;
    }
    printf("Enter element to insert: ");
    scanf("%d", &n);
    if(front == -1)
    {
        front = rear = 0;
        dequeue[front] = n;
    }
}
```

```

    }
    else
    {
        if(front == 0)
            front = MAX - 1;
        else
            front--;
        dequeue[front] = n;
    }
}

void delete_front()
{
    if(front == -1)
    {
        printf("\nQueue is empty.\n");
        return;
    }
    printf("Deleted element: %d\n", dequeue[front]);
    if(front == rear) front = rear = -1;
    else front = (front + 1) % MAX;
}

void delete_rear()
{
    if(front == -1)
    {
        printf("\nQueue is empty.\n");
        return;
    }
    printf("Deleted element: %d\n", dequeue[rear]);
    if(front == rear)
        front = rear = -1;
    else
    {
        if(rear == 0)
            rear = MAX - 1;
        else
            rear--;
    }
}

void display()
{
    if(front == -1)
    {
        printf("\nQueue is empty.\n");
        return;
    }
    printf("\nDequeue: ");
    for(i = front; i != rear; i = (i + 1) % MAX)
        printf("%d\t", dequeue[i]);
    printf("%d\t", dequeue[i]);
}

void main()
{
    int ch;

```

```
do
{
    printf("\nMENU\n");
    printf("1. Insert from front\n2. Insert from rear\n3. Delete from front\n");
    printf("4. Delete from rear\n5. Display\n6. Exit");
    printf("\nEnter choice: ");
    scanf("%d", &ch);
    switch(ch)
    {
        case 1: insert_front();
                break;
        case 2: insert_rear();
                break;
        case 3: delete_front();
                break;
        case 4: delete_rear();
                break;
        case 5: display();
                break;
    }
}while(ch >= 1 && ch <= 5);
}
```

OUTPUT

```
D:\Study\Lab\Data-Structures-Programs>cd "d:\Study\Lab\Data-Structures-Programs\Day 5\Day 5\"dequeue
```

```
MENU
```

1. Insert from front
2. Insert from rear
3. Delete from front
4. Delete from rear
5. Display
6. Exit

```
Enter choice: 1
```

```
Enter element to insert: 1
```

```
MENU
```

1. Insert from front
2. Insert from rear
3. Delete from front
4. Delete from rear
5. Display
6. Exit

```
Enter choice: 1
```

```
Enter element to insert: 0
```

```
MENU
```

1. Insert from front
2. Insert from rear
3. Delete from front
4. Delete from rear
5. Display
6. Exit

```
Enter choice: 2
```

```
Enter element to insert: 2
```

```
MENU
```

1. Insert from front
2. Insert from rear
3. Delete from front
4. Delete from rear
5. Display
6. Exit

```
Enter choice: 3
```

```
Deleted element: 0
```

MENU

1. Insert from front
2. Insert from rear
3. Delete from front
4. Delete from rear
5. Display
6. Exit

Enter choice: 5

Dequeue: 1 2

MENU

1. Insert from front
2. Insert from rear
3. Delete from front
4. Delete from rear
5. Display
6. Exit

Enter choice: 4

Deleted element: 2

MENU

1. Insert from front
2. Insert from rear
3. Delete from front
4. Delete from rear
5. Display
6. Exit

Enter choice: 5

Dequeue: 1

MENU

1. Insert from front
2. Insert from rear
3. Delete from front
4. Delete from rear
5. Display
6. Exit

Enter choice: 6

d:\Study\Lab\Data-Structures-Programs\Day 5>