# WEEK 3

## 1. Queue Implementation

```
#include <stdio.h>
int q[50], rear = -1, front = -1, size;
void enqueue();
void dequeue();
void display();
void main()
{
    int ch;
    printf("Enter the size of queue:");
    scanf("%d", &size);
    printf("Enter choice:");
    printf("Press 1. insert, 2. delete, 3. Display
            and 4. Exit \n");
    while(ch!=4)
    {
        printf("Enter choice:");
        scanf("%d", &ch);
        switch(ch)
        {
            case 1:
                enqueue();
                break;
            case 2:
                dequeue();
                break;
            case 3:
                display();
                break;
    } }
```

```
        printf("Exited");
}

void enqueue()
{
    int item;
    if(rear == size-1)
        printf("Queue is full\n");
    else
    {   if(front == -1)
            front = 0;
        printf("Insert an element: ");
        scanf("%d", &item);
        rear += 1;
        q[rear] = item;
    }
}

void dequeue()
{
    if (front == -1 || front > rear)
        printf("Queue is empty\n");
    else
    {
        printf("Deleted element is: %d\n", q[front]);
        front += 1;
    }
}

void display()
{
    int i;
```

```
if (front == -1)
        printf("Queue is empty");
else
{
        printf("Queue is: \n");
        for(int i=front; i<=rear; i++)
                printf("%d\t", q[i]);
        printf("\n");
}
}
```

OUTPUT -

```
Press - 1. Insert, 2. delete, 3. Display and 4. Exit
Enter choice: 2
Queue is empty
Enter choice: 1
Insert an element: 6
Enter choice: 1
Insert an element: 8
Enter choice: 1
Insert an element: 5
Enter choice: 2
Deleted element is: 6
Enter choice: 3
Queue is:
8    5
Enter choice: 4
Exited
```

11/1/24

WEEK -3

2. Circular Queue Implementation

```
#include <stdio.h>
int q[50], front = -1, rear = -1, size;
void display();
void enqueue();
void dequeue();
void main()
{
    int ch;
    printf("Enter no. of elements :");
    scanf("%d", &size);
    while(ch != 4)
    {
        printf("1. Insert 2. Delete 3. Display 4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &ch);
        switch(ch)
        {
            case 1:
                enqueue();
                break;
            case 2:
                dequeue();
                break;
            case 3:
                display();
                break;
        }
    }
    printf("Exited");
}
```

```
void enqueue()
{
    int item;
    if ((front == rear+1) || (front == 0 && rear == size-1))
        printf("Queue is full\n");
    else
    {
        if (front == -1)
            front = 0;
        printf("Enter the element : ");
        scanf("%d", &item);
        rear = (rear+1) % size;
        q[rear] = item;
    }
}

void dequeue()
{
    int ele;
    if (front == -1)
        printf("Queue is empty \n");
    else
    {   ele = q[front];
        if (front == rear)
        {   front = -1;
            rear = -1;
        }
        else
        {   front = (front+1) % size;
        printf("Deleted element = %d \n", ele);
}   }
```

```
void display()
{
    int i;
    if(front == -1)
        printf("Queue is empty");
    else
    {   printf("Front = %d \t", front);
        printf("Rear = %d \t", rear);
        printf("Queue is:");
        for(int i = front ; i != rear; i = (i+1)%size)
            printf("%d", q[i]);
        printf("%d\n", q[i]);
    }
}
```

OUTPUT :

```
Enter no. of elements : 5
1. Insert  2. Delete  3. Display  4. Exit
Enter your choice : 1
Enter element : 4
Enter your choice : 1
Enter element : 2
Enter your choice : 1
Enter element : 3
Enter your choice : 1
Enter element : 5
Enter your choice : 1
Enter element : 6
Enter your choice : 1
Queue is full
Enter your choice : 2
```

Deleted element = 4
Enter your choice : 2
Deleted element = 2
Enter your choice : 1
Enter element = 8
Enter your choice : 3
Front = 2 Rear = 0
Queue : 3568
Enter your choice : 4
Exited.