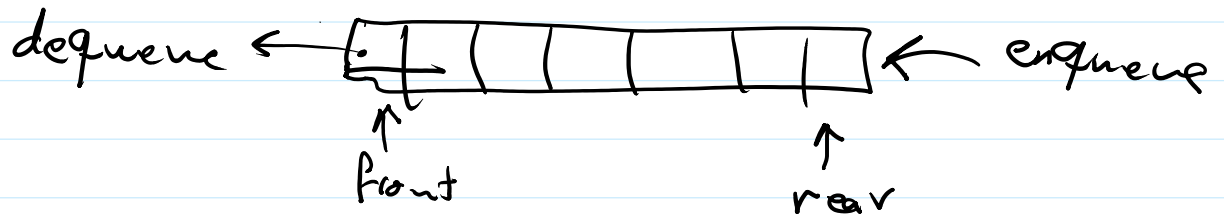


# Queue

Wednesday, March 24, 2021 11:37 AM

Queue! is a list, whereas the insertion of new node is performed always at the end of queue, while the deletions are performed at the opposite side.

Queue! First In First Out : FIFO



\* Applications on queue!

- 1- printers
- 2- OS and computer HW.
- 3- calling centers.
- 4- Reg. sys.
- 5- Reg. sys.

\* Implementation of Queue : 1- Pointer Imp.  
2- Array Imp.

```
struct Queue {  
    int Data[SIZE];  
    int front;  
    int rear;  
}; Q;
```

Initialization of Queue..

```
Q.front = SIZE - 1;
```

```
Q.rear = SIZE - 1;
```

```
int isEmpty(Queue Q) {
```

```
    return (Q.front == Q.rear);
```

```
}
```

```
void enqueue(Queue Q, int x) {
```

```
    int temp = Q.rear;
```

```
    if (Q.rear == SIZE - 1)
```

```
}
```

```

int temp = Q.rear;
if (Q.rear == size - 1)
    Q.rear = 0;
else
    ++Q.rear;
if (Q.rear == Q.Front) {
    printf("Error: Queue is Full");
    Q.rear = temp;
}
else
    Q.Data[Q.rear] = x;
}

int dequeue (Queue Q) {
    if (!isEmpty(Q)) {
        if (Q.Front == size - 1)
            Q.Front = 0;
        else
            ++Q.Front;
        return Q.Data[Q.Front];
    }
    else {
        printf("Error: Queue is Empty");
        return -1;
    }
}

```