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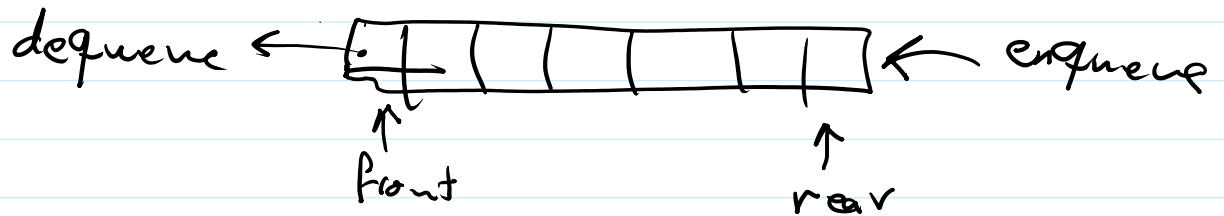
Queue: is a list, whereas the insertion of new node is performed always at the end of queue, while the deletions are performed at the opposite side.

Queue: First In First Out : FIFO

dequeue ← [............] ← enqueue

front   rear

*Applications on Queue:
1- pointers
2- OS and computer HW.
3 - Calling centers.
4 - Reg. sys.
5 - Res. Sys.
*Implementation of Queue : 1- Pointer Imp.
                            2- Array Imp.

```
struct Queue{                 ⎫
   int Data [SIZE];           ⎬  Initialization of Que.
   int front;                 ⎭
   int rear;
 } Q;
 Q. front = SIZE-1;
 Q. front = 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;
    }
}
```