```python
class Queue:
    # max_size: size of Q
    # Q: Array
    def __init__(self, max_size):
        self.max_size = max_size
        self.Q = [0] * max_size
        self.num = 0
        self.first = 0

    def enqueue(self, item):
        if self.num >= self.max_size:
            raise Exception("Queue overflow")
        self.Q[(self.num + self.first) % self.max_size] = item
        self.num += 1


    def dequeue(self):
        if self.num == 0:
            raise Exception("Queue empty")
        item = self.Q[self.first]
        self.first = (self.first + 1) % self.max_size
        self.num -= 1
        return item


    def front(self):
        if self.num == 0:
            raise Exception("Queue empty")
        return self.Q[self.first]

    def is_empty(self):
        return self.num == 0
```

```python
    def is_empty(self):
        return self.num == 0

    def size(self):
        return self.num

    def is_full(self):
        return self.num >= self.max_size

    def dequeue_i(self , i):
        if self.num == 0:
            raise Exception("Queue empty")
        item=self.Q[i]
        del self.Q[i]
        index=[(self.num + self.first) % self.max_size]
        self.Q.insert(0,index)
        self.num-=1
        return item
```