```c
#include <semaphore.h>
#include <pthread.h>
#include <stdlib.h>
#include "ProdCons.h"

Queue * Queue_init(int N){
    Queue * q = malloc (sizeof(Queue) );
    q->size = N;
    q->buffer = malloc( N * sizeof(void*) );
    sem_init(&q->empty, 0, N);
    sem_init(&q->full, 0, 0);

    return q;
}

void Queue_destroy(Queue * q){
    free ( (*q).buffer);
    sem_destroy(&q->empty);
    sem_destroy(&q->full);
    free(q);
}

void Queue_insert(Queue * q, void * item){
    sem_wait(&(q->empty));
    q->buffer[q->insert_pos] = item;
    q->insert_pos =  (q->insert_pos + 1) % q->size;
    sem_post(&(q->full));
}

void * Queue_extract(Queue * q){
    void * item;
    sem_wait(&(q->full));
    item = q->buffer[q->extract_pos];
    q->extract_pos =  (q->extract_pos + 1) % q->size;
    sem_post(&(q->empty));
    return item;
}
```