```
/* Implementation of a simple circular queue using a static array */
#include <stdio.h>
#include <stdlib.h>
#include "queue.h"

/* create the queue data structure and initialize it */
queue *queue_init(int n)
{
        queue *q = (queue *)malloc(sizeof(queue));
        q->size = n;
        q->buffer = malloc(sizeof(job *) * n);
        q->start = 0;
        q->end = 0;
        q->count = 0;

        return q;
}

/* insert an item into the queue, update the pointers and count, and
   return the no. of items in the queue (-1 if queue is null or full) */
int queue_insert(queue *q, job *item)
{
        if ((q == NULL) || (q->count == q->size))
                return -1;

        q->buffer[q->end % q->size] = item;
        q->end = (q->end + 1) % q->size;
        q->count++;

        return q->count;
}

/* delete an item from the queue, update the pointers and count, and
   return the item deleted (-1 if queue is null or empty) */
job *queue_delete(queue *q)
{
        if ((q == NULL) || (q->count == 0))
                return NULL;

        void *x = q->buffer[q->start];
        q->start = (q->start + 1) % q->size;
        q->count--;

        return x;
}

/* delete the queue data structure */
void queue_destroy(queue *q)
{
        free(q->buffer);
        free(q);
}
```