Worksheet 18: Linked List Queue, pointer to Tail

```
struct link {
	TYPE value;
	struct link * next;
};

struct listQueue {
	struct link *firstLink;
	struct link *lastLink;
};

void listQueueInit (struct listQueue *q) {
	struct link *lnk = (struct link *) malloc(sizeof(struct link));
	assert(lnk != 0); /* lnk is the sentinel */
	lnk->next = 0;
	q->firstLink = q->lastLink = lnk;
}

void listQueueAddBack (struct listQueue *q, TYPE e) {
	struct link *lnk = (struct link *) malloc(sizeof(struct link));
	lnk->value = e;
	lnk->next = 0;
	q->lastLink->next = lnk;	//the “next” value of the link lastLink currently points to =
	lnk
	q->lastLink = lnk;		//lastLink now points to lnk
}


TYPE listQueueFront (struct listQueue *q) {
	assert(q->firstLink != q->lastLink);
	return q->firstLink->next->value;
}


void listQueueRemoveFront (struct listQueue *q) {
	assert(q->firstLink != q->lastLink);
	struct link *tempLink = q->firstLink->next;
	q->firstLink->next = tempLink->next;
	free(tempLink);
}
```

```
int listQueueIsEmpty (struct listQueue *q) {
	assert(q!=0);
	if (q->firstLink == q->lastLink)
		return 1;
	else
		return 0;
}
```