# Stack interface

**// typedef \_\_\_\_\_\_\*stack\_t;**

**bool stack\_empty(stack\_t S) /\* O(1) \*/**

**/\*@requires S != NULL; @\*/ ;**

**stack\_t stack\_new() /\* O(1) \*/**

**/\*@ensures \result != NULL; @\*/**

**/\*@ensures stack\_empty(\result); @\*/ ;**

**void push(stack\_t S, string x) /\* O(1) \*/**

**/\*@requires S != NULL; @\*/ ;**

**string pop(stack\_t S) /\* O(1) \*/**

**/\*@requires S != NULL; @\*/**

**/\*@requires !stack\_empty(S); @\*/ ;**

# Queue interface

**// typedef \_\_\_\_\_\_\* queue\_t;**

**bool queue\_empty(queue\_t Q) /\* O(1) \*/**

**/\*@requires Q != NULL; @\*/ ;**

**queue\_t queue\_new() /\* O(1) \*/**

**/\*@ensures \result != NULL; @\*/**

**/\*@ensures queue\_empty(\result); @\*/ ;**

**void enq(queue\_t Q, string e) /\* O(1) \*/**

**/\*@requires Q != NULL; @\*/ ;**

**string deq(queue\_t Q) /\* O(1) \*/**

**/\*@requires Q != NULL; @\*/**

**/\*@requires !queue\_empty(Q); @\*/ ;**