Exercise 1.
Implement a stack in array $S[1..n]$. Remember to detect overflow and underflow errors.

Exercise 2. (Cormen 10.1-2)
Implement two stacks in one array $A[1..n]$ in such a way that neither stack overflows unless the total number of elements in both stacks together is $n$. The PUSH and POP operations should run in $O(1)$ time.

Exercise 3.
Implement a queue in array $Q[1..n]$. Remember to detect overflow and underflow errors.

Exercise 4. (Cormen 10.1-5)
Whereas a stack allows insertion and deletion of elements at only one end, and a queue allows insertion at one end and deletion at the other end, a deque (double-ended queue) allows insertion and deletion at both ends. Write four $O(1)$-time procedures to insert elements into and delete elements from both ends of a deque constructed from an array.

Exercise 5.
Implement a singly linked list (searching for a given key, inserting a node at the beginning of the list and deleting a given node).

Exercise 6.
Write a procedure that inserts node $x$ after a given node $z$ in
(a) a singly linked list,
(b) a doubly linked list.

Exercise 7.
Write a procedure that inserts node $x$ before a given node $z$ in
(a) a doubly linked list,
(b) a singly linked list.