Exercise Sheet 2

1. Perform the following sequence of operations on a queue, which initially is empty. Say after each operation what is the content of the queue.

   - enqueue(5)
   - enqueue(3)
   - dequeue()
   - enqueue(2)
   - enqueue(8)
   - dequeue()
   - dequeue()
   - enqueue(9)
   - enqueue(1)

2. Provide pseudocode descriptions for the operations size(), isEmpty() and head() of the abstract data type for queues.

3. Let us have two (unbounded) stacks at our disposal, with the operations as specified in the abstract data type for stacks. Use them to implement the operations enqueue and dequeue of the abstract data type for queues.
   Do this in such a way that the operation enqueue is in $O(1)$, and the operation dequeue on average is in $O(1)$.

4. Let us have two (unbounded) queues at our disposal, with the operations as specified in the abstract data type for queues. Use them to implement the operations push and pop of the abstract data type for stacks.
   Analyze the time complexity of push and pop in terms of $O$.

5. Provide pseudocode descriptions for the operations inList(v) and insertBefore(v,d) on singly-linked lists, as well as on doubly-linked lists.

6. Give a $\Theta(n)$-time non-recursive procedure that reverses a singly-linked list of $n$ elements. The procedure should use no more than constant storage space beyond that needed for the list itself.