1. Draw a picture of the node based stack after the following operations:

```
1    s = NONE_NODE
2    s = push(s, "a")
3    s = push(s, "b")
4    s = push(s, "c")
5    s = pop(s)
6    s = push(s, "d")
7    s = push(s, "e")
```

2. Draw a picture of the node based queue after the following operations:

```
1    q = mkQueue()
2    enqueue(q, "a")
3    enqueue(q, "b")
4    dequeue(q)
5    enqueue(q, "c")
6    enqueue(q, "d")
7    enqueue(q, "e")
```

3. Give 2 examples of a stack and a queue in real life.

4. Give the Big-O complexity for each (using the lecture implementation of stacks and queues)
   (a) Pushing a new element onto a stack
   (b) Getting the size of a stack
   (c) Telling if a stack is empty (or not)
   (d) Removing the front element from a queue
   (e) Reversing the elements in a queue

5. Show the output for each call to `isinstance`.

   (a) `>>> isinstance(10, int)`

   (b) `>>> isinstance('2', str)`

   (c) `>>> isinstance(10, float)`

   (d) `>>> isinstance(True, False)`

   (e) `>>> isinstance(NONE_NODE, Node)`

6. What does the lecture code do if you try to `pop` an empty stack?

7. Here is the node definition from lecture:

```
1     class NoneNode():
2         __slots__ = ()
3
4     NONE_NODE = NoneNode()
5
6     class Node():
7         __slots__ = ( 'data', 'next' )
8
9     def mkNode(dataVal, nextVal):
10        """Create and return a newly initialized Node object"""
11        node = Node()
12        node.data = dataVal;
13        node.next = nextVal;
14        return node;
```

   Write the function, `pop`, for a stack. It removes the top element (and does not return it). If the stack is empty, `emptyStack`, it should raise an exception.

8. Here is the queue definition from lecture:

```
1     class Queue():
2         __slots__ = ( 'front', 'back', 'size' )
3
4     def mkQueue():
5         q = Queue()
6         q.front = NONE_NODE
7         q.back = NONE_NODE
8         q.size = 0
9         return q
```

   Write the function, `dequeue` for a queue. It should remove the front element from the queue. If the queue is empty, `emptyQueue`, it should raise an exception.