1. If a Vector ADT is implemented using an array, then, an at(i) operation takes time \( O(1) \) and an insert(i, e) operation takes time \( O(n) \).

2. If a Vector ADT is implemented using a doubly-linked list, then an at(i) operation takes time \( O(n) \) and an insert(i, e) operation takes time \( O(n) \).

3. If a List ADT is implemented using an array, then a begin() operation takes time \( O(1) \) and an insert(p, e) operation takes time \( O(n) \).

4. If a List ADT is implemented using a doubly-linked list, then a begin() operation takes time \( O(1) \) and an insert(p, e) operation takes time \( O(1) \).

5. The Sequence ADT generalizes the Vector ADT and the List ADT, and provides access to elements using both atIndex(i), indices and indexOf(p), positions.

6. True or False: If a Sequence ADT is implemented using a circular array, then all operations can be done in \( O(1) \) time.

7. True or False: If a Sequence ADT is implemented using a doubly-linked list, then all operations can be done in \( O(1) \) time.