Week 3 Quiz 1

1. What is one benefit of using linked lists over arrays?
   - Linked lists have fixed memory allocation.
   - Linked lists automatically prevent memory leaks.
   - Linked lists allow flexible insertion and deletion of elements.
   - Linked lists are easier to sort than arrays.

   One of the main benefits of linked lists is their flexibility in inserting and deleting elements without the need for shifting other elements, unlike in arrays.

2. Select each of the following statements about linked lists that is true.
   - Linked lists are more efficient than arrays when inserting a new item, especially to a location in the middle of the linked list

     Inserting an item to the middle of an array requires shifting over every following item, leading to significant inefficiencies. Insertion with a linked list, on the other hand, requires traversal to the desired location, but then only requires updating the pointers of the nodes at the point of insertion.
   - The items in a linked list must be sorted for the linked list to operate correctly
   - Linked lists are more efficient than arrays for accessing individual items
   - Linked lists require additional memory to store a pointer with each item that points to the next item

     The pointer included with each element in the node of a linked list does in fact require additional memory.

3. What is a key advantage of using arrays in Python?
   - Arrays allow quick access and modification if you know the index of the item to be accessed
   - Arrays automatically sort the data they contain
   - Arrays can store even billions of items without performance or memory concerns
   - Arrays are an efficient data structure when you will frequently be inserting and deleting items

     One of the main advantages of arrays is their ability to allow quick access and modification of elements when the index is known, making them efficient for certain operations.

4. Select each of the following true statements on working with an LLM to choose and implement data structures.
   - Assigning the LLM an expert role can allow you to profile the data structure decisions you make from a particular perspective
   - LLMs can generate code to implement a certain data structure in the context of your project

- Only beginner programmers are likely to benefit from working with an LLM to implement data structures in their programs
- LLMs can provide generic information on data structures but can't relate that information to the context of your specific projects

5. Consider the following code that uses the Node and LinkedList classes covered in the lectures. What will be printed when this code is run?

```python
import threading

class Node:
    def __init__(self, data):
        self.data = data
        self.next = None

class LinkedList:
    def __init__(self, max_size=None):
        self.head = None
        self.size = 0
        self.max_size = max_size
        self.lock = threading.Lock()

    def append(self, data):
        # Validate input data
        if len(data) > 1000:
            raise ValueError("Data size exceeds maximum limit")
        with self.lock:
            if self.max_size is not None and self.size >= self.max_size:
                raise ValueError("Linked list is full")
            new_node = Node(data)
            if self.head is None:
                self.head = new_node
            else:
                last = self.head
                while last.next:
                    last = last.next
                last.next = new_node
            self.size += 1

    def print_list(self):
        current = self.head
        while current:
            print(current.data, end=" ")
            current = current.next

mylist = LinkedList()
mylist.append("X")
mylist.append("X")
```

- XYZ
- XXYZ
- ZYX
- ZYXX