Question 1

Firstly, we created our own vector and linked list class. We made a templated class for the vector so we can take in any type like a string or integer. Variables “curr,” “storage,” and “ptr” were initialized so we can use them for the class functions we will make. The push function adds a value to the end of the vector. In order to do this, we have to increase the storage size, copy all the values from the original vector into one with the bigger storage, then add the value after all values are copied. Insertion lets us put a value into a specific position, we do this by adding space to the storage of the vector, copying values of the vector into the bigger vector up until the point of the position we want, then we put the value into position, and then copy the rest of the values in one position higher than their original. Our deletion function gets rid of a value at a given position. This is done by decreasing the storage space, copying the values into the new smaller vector until the position we called, then we start from that position but add values of the elements that come after our position. We also made simple functions like get (which tells you the position), pop (which just lessens the storage size to get rid of the last value), size (which returns the size of the vector), etc. Then we incorporated functions to fill vectors with numbers, and one with strings. This is done by using the random function which generates random values. We use a for loop to iterate through the positions of the vectors and fill them in. In main, we test the timing starting from when we call the function.

The linked list class first introduces a struct for building a node (data and a pointer to the next node). In the linked list class, we initialize a head and tail. The function firstnode takes in the head, tail, and value and creates a new node, filling its data with the value, and initializing the pointer to the next node. We set this node to be the head and tail. Another function “isEmpty” will take in the head node, and simply checks if it is empty. Insert will firstly call the isEmpty function to make sure there is a node at all. If not, it will create a new node and insert the value to it. In main we used a for loop to fill a list with nodes of random integers and another with random strings. We time it from the start of creating the list to the end of filling it and printing it.