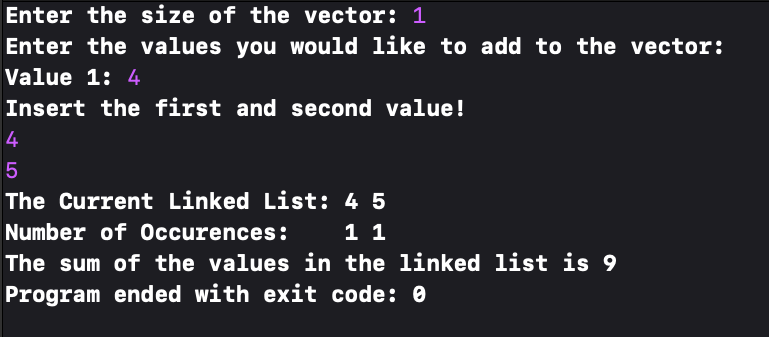
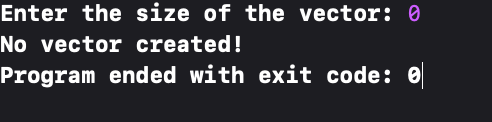
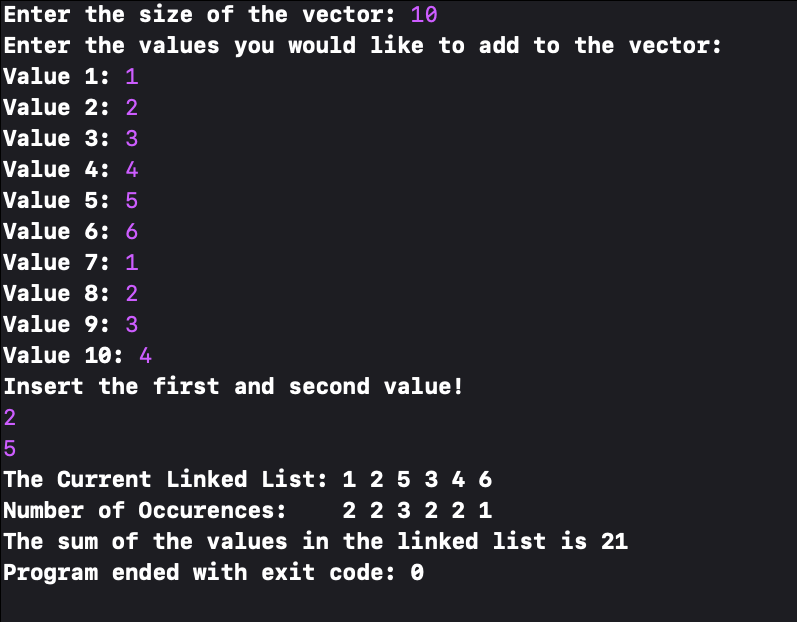
Kareem Elnaghy 900223587

**Assignment 4 Report**



Note:

* The linked list class and the node class were all based on the algorithms provided in Dr. Howaida’s Slides and what was showed in the hidden labs.
* It wasn’t specified if we shouldn’t use built in functions for vectors, so I used a few functions such as the push\_back function and the insert function which were the only ones I found to be necessary.

In the case that 0 was entered for the size of the vector, the program would display a message that no vector was created and that it would exit the program. The if condition that executes this command is present in the main file.

In this case, the size of the vector is 1, therefore there will be only one value in the vector before the insert after function is called. Therefore, when creating the linked list, there is a specific condition for when there is one element. This checks if the size of the vector is 1 and if it is, it will use the push\_back function to add the second value instead of going through the else statement where we iterate through the vector, comparing each element to insert the second value. The program then displays the linked list and the number of occurrences and the sum of the values in the list.

This is a normal instance with the size of the vector 10. Numbers were then filled in the vector and for every occurrence of 2 in the vector, 5 was added next to it. The program then displays the list without any duplicates and displays the occurrences of each number right under the respective number. E.g. 2 occurrences for 1 and 3 occurrences for 5. The program then displays the sum of all the values in the linked list. 1+2+5+3+4+6 = 21. This is done by the function sumList which traverses through the list and adds each value in the node to a variable sum that is continuously being updated. At the end of the function, the sum is returned and displayed in the main.

Different scenarios and different outputs: