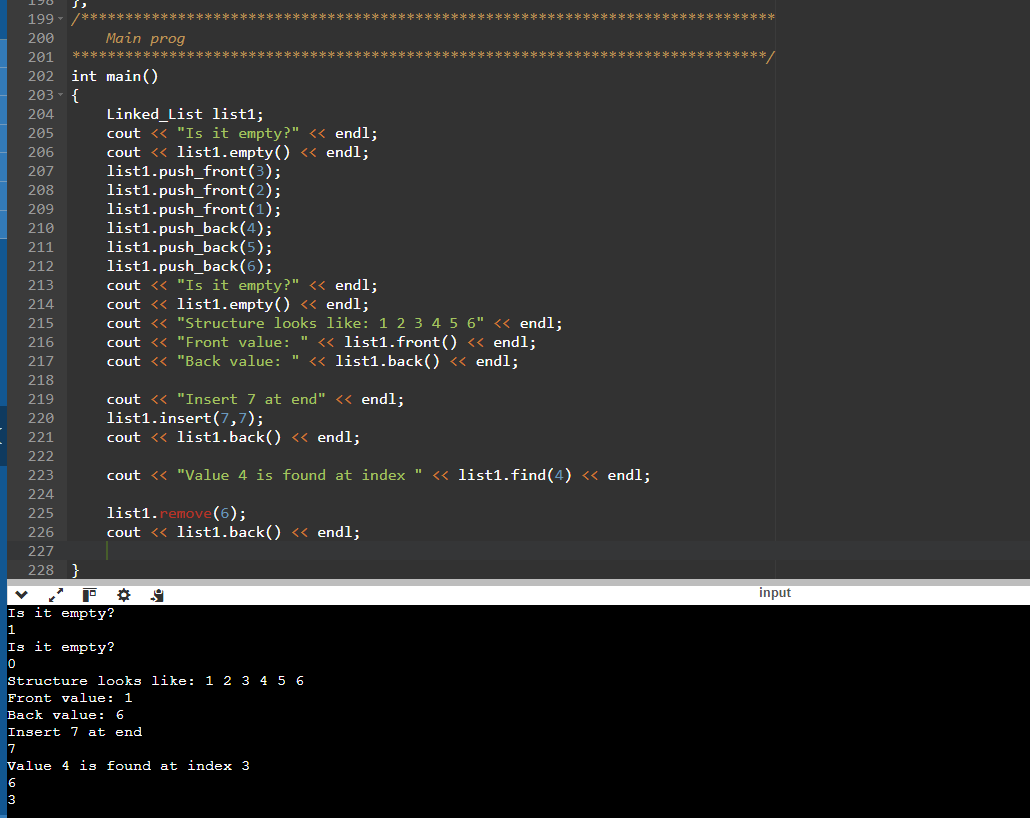
Question 1

Code for question 1 is using a list of Node objects to mimic a single linked list. Each created node has an element of type integer and a pointer pointing to the next node. The head and tail node are also saved within the object.

Possible functions and their purpose listed below

Push\_Front - Add value to the beginning of the linked\_list  
Pop\_Front – Remove and return first value  
Push\_Back – Add value to the end of the linked\_list  
Pop\_Back – Remove and return last value  
Front – Return front value  
Back – Return last value  
Empty – True false if list is empty (1 or 0)  
Insert(insert index, insert element) – Insert given value at given index  
Remove(remove index) – Remove value at given index



Question 2

Code for question 2 is creating an object called Stack that is mimicking a Stack data structure using a vector. The end of the vector is being treated as the top and all the functions are based around the Last in First Out idea.

Possible functions and their purpose listed below

Push(add int) – Add a value to the top of the stack  
Empty – True false if stack is empty (1 or 0)  
Size – Return size of the stack  
Pop – Remove the top stack  
Top – Return the value at the top of the stack  
Average – Return average value of all elements in the stack

