The programme is made to use C++ to implement a linked list data structure. Three files make up the programme: main.cpp and LinkedLists.cpp and LinkedLists.h. The implementation of the linked list class, which includes the required member functions to add a node, remove a node, print all linked list elements, generate a linked list based on a vector's elements, and return the sum of all nodes, can be found in the LinkedLists.cpp file.

The class is defined in LinkedLists.h, which includes a struct called Node that contains two fields: value and count. The count field shows how many times that number appears in the vector, whereas the value field shows its value in the node.

The constructor and destructor of the LinkedList class, which initialise the list's head to null and remove all nodes from the list, respectively. With a value and count both set to 1, the addNode function adds a new node to the list's head. The removeNode function eliminates a node from the list whose value is provided. The list's elements and counts are printed by the printList function. The createList function builds a linked list based on the elements of an integer vector. If a value appears more than once in the vector, the count of that node in the LinkedList reflects the number of occurrences. After the first instance of the node with the specified value, the function insertAfter adds a new node with the specified value. The function performs nothing if a node without the initial value cannot be found.

Finally, the sumNodes function returns the sum of all nodes in the linked list.

The main.cpp code shows how the linked list class works by allowing the user to input an integer vector, add or remove values from the list, print the list, create a linked list based on the elements of the vector, and return the sum of all the nodes in the list.

Overall, the code is cleanly organised, simple to comprehend, and it satisfies the task's objectives. The implementation of the createList and sumNodes functions is made easier by the creative usage of a struct to hold the count of each value. Because it only keeps the count of each value in the linked list once, the code is effective and does not waste memory.

Output:  
