1) This is a function that takes 3 parameters:

A vector of type integer

A number

Another number

This function is made to insert the second number after the occurrence of the first number in the vector.

```
void inserAfer(vector<int> v, int firstnum, int secnum) {
    for (int i = 0; i < v.size(); i++) {
        if (v[i] == firstnum) {
            v[i + 1] = secnum;
        }
    }
}
int main() {
```

2) in the main function I wrote a code that asks the user to input a number that will then be added to a vector, then I asked the user if he want to add another number or not and I validated the user answer to y or n all of this was done In a loop that ends once the user types n

```
12  int main() {
13      vector<int> v;
14      int num;
15      cout << "please enter a number";
16      cin >> num;
17      v.push_back(num);
18      bool flag = true;
19      char answer;
20      while (flag == true) {
21          cout << "do you want to enter another number" << endl;
22          cin >> answer;
23          if ((answer != 'y' )&& (answer!='n')) {
24              cout << "wrong answer";
25          }
26          else {
27              if (answer == 'n') {
28                  flag = false;
29              }
30              else {
31                  if (answer == 'y') {
32                      cout << "please enter a number";
33                      cin >> num;
34                      v.push_back(num);
35                  }
36              }
37          }
38      }
```

3) Then I created a new linked list using the class I created, and I called the convert function I did to convert a vector to a linked list.

```
    lists l;
    (l.converttolist(v)).printlist();
    cout << l.sumofnodes((l.converttolist(v)));


}
```

4) Speaking about the linked list class:
firstly I formed a struct called node which contained the following attributes.

```
class lists
{
    struct node {
        int value;
        int counter=1;
        node* next;
        node* prev;
    };
```

Then I wrote a function to add a node to the list and a function to delete a node

```
void addnode(int v) {
    node* n = new node;
    n->value = v;
    n->next = NULL;
    n->prev = NULL;
    if (head == NULL) {
        head = n;
        tail = n;
    }
    else {
        tail->next = n;
        n->prev = tail;
        tail = n;

    }

}

void deletenode(int v) {
    node* deleteptr = NULL;
    temp = head;
    current = head;
    while (current != NULL && current->value != v) {
        temp = current;
        current = current->next;
    }
    if (current == NULL) {
        cout << "the value was not in the list";
    }
    else {
        deleteptr = current;
        current = current->next;
        temp->next = current;
        delete deleteptr;
        cout << "the value is deleted";

    }

}
```

Then I wrote a function to print the linked list

```
void printlist() {
    node* trav = head;
    while (trav != NULL) {
        cout << trav->value << endl;
        trav = trav->next;
    }
}
```

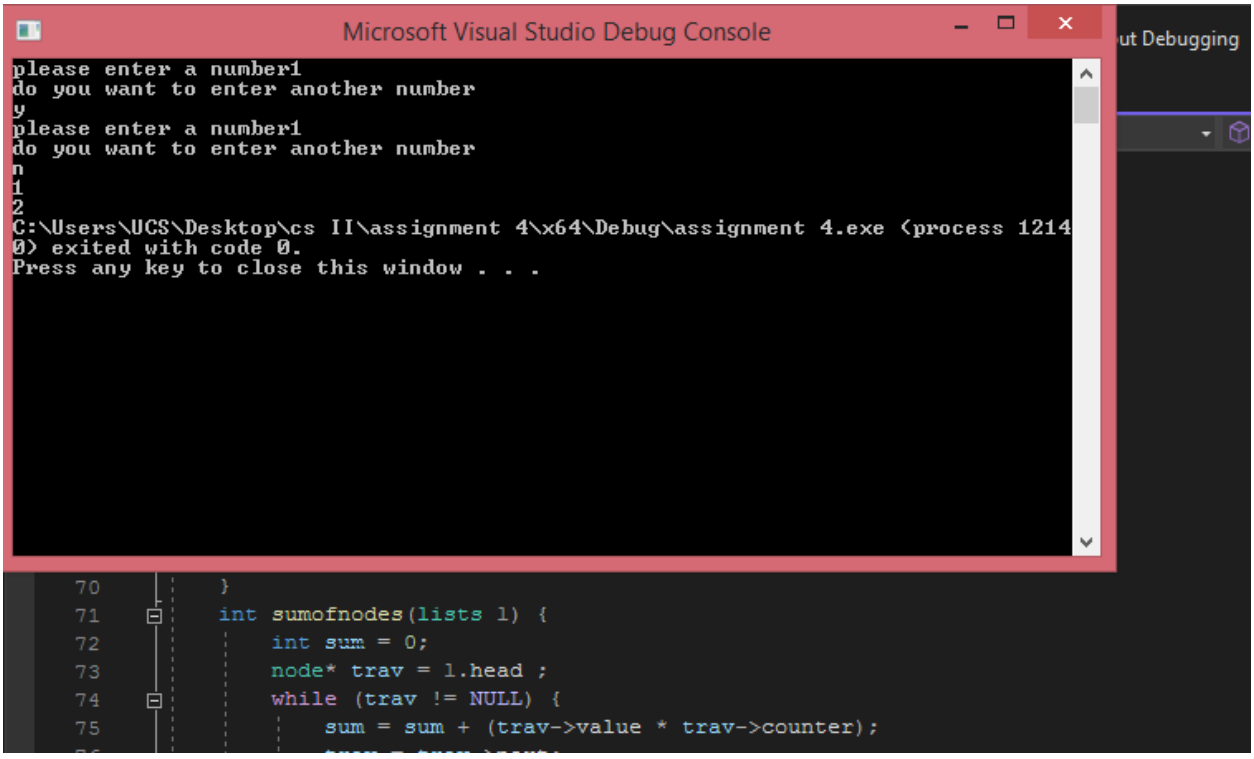
And a function to calculate the sum of the nodes

```
int sumofnodes(lists l) {
    int sum = 0;
    node* trav = l.head ;
    while (trav != NULL) {
        sum = sum + (trav->value * trav->counter);
        trav = trav->next;
    }return sum;
}
```

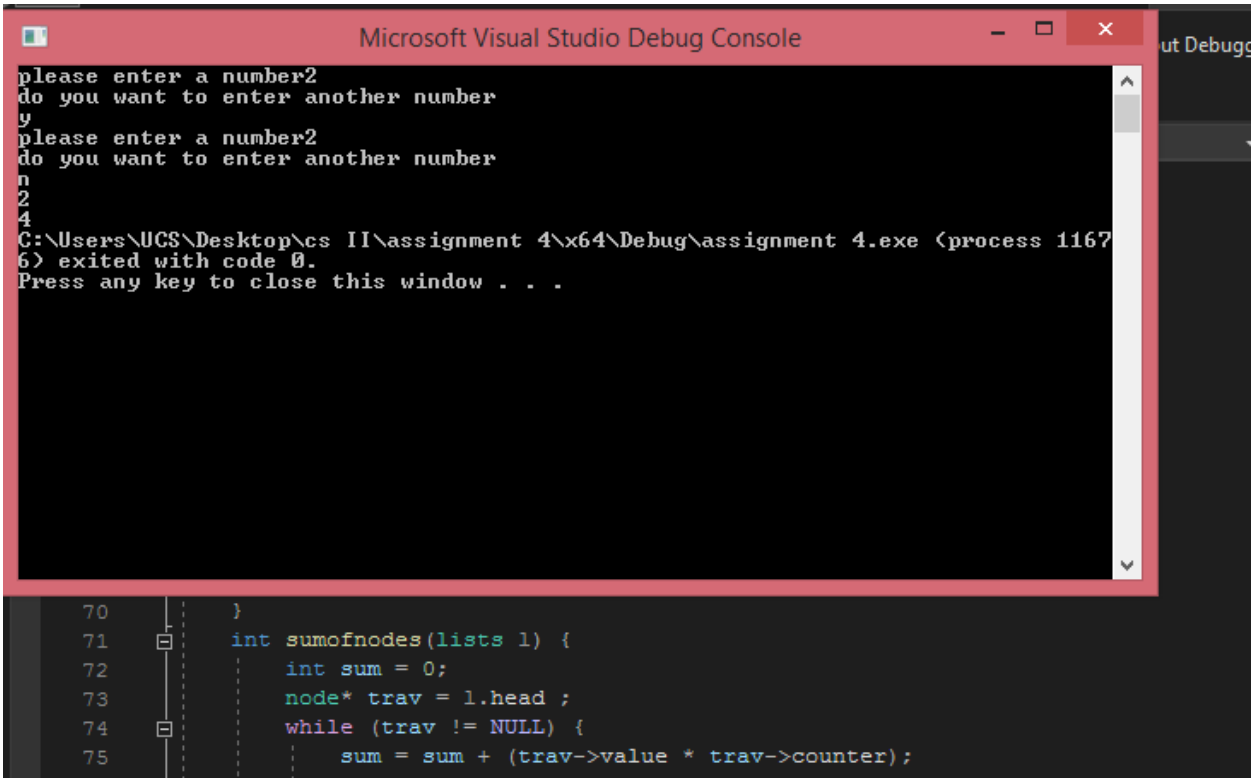
Results

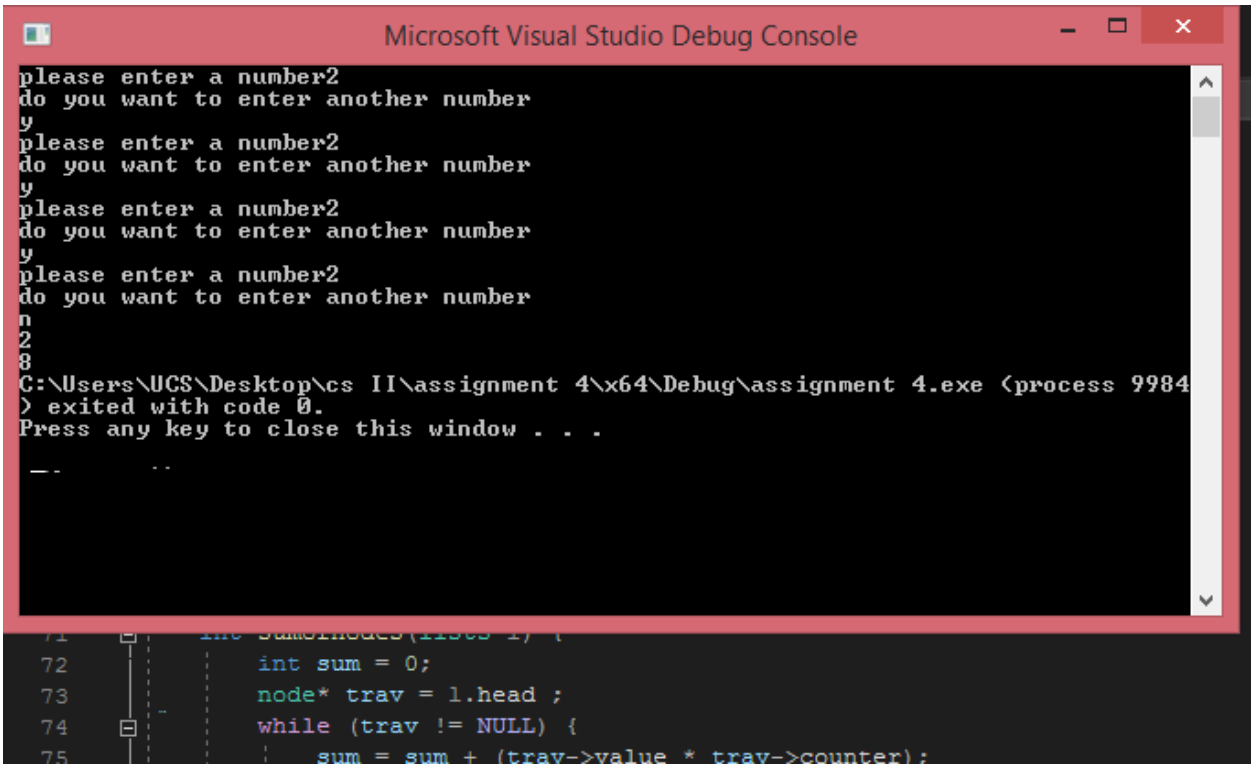
here is the result when the user entered two identical numbers.

The number was shown one time which indicates that the one node only was added and the sum was 2 which indicated that the number of occurrences was reflected in the counter



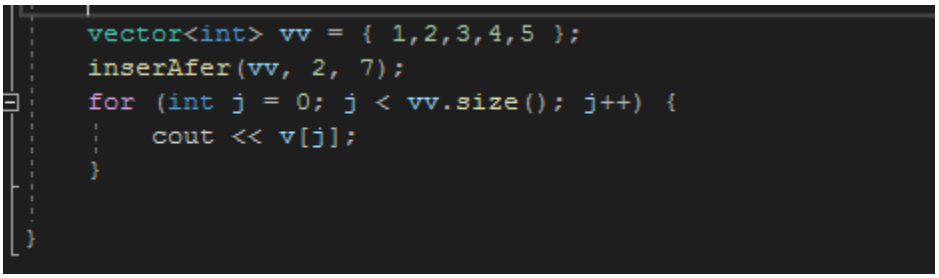
Here is another example





Unfortunately the code was not responding when I try to input numbers that are not similar to each other

Here is the use of the insert function in the main



```
    vector<int> vv = { 1,2,3,4,5 };
    inserAfer(vv, 2, 7);
    for (int j = 0; j < vv.size(); j++) {
        cout << v[j];
    }

}
```