

Linked List: Number + 1  
450  
Input: 4 -> 5 -> 0  
Output: 4 -> 5 -> 1

Ans

```
ListNode*next;
Int addOneutil(ListNode*node) {
s
if(node == null) return 1;
Int carry = addoneutil(node->next);
Int sum = node->val+carry;
node->val= sum%10;
Return sum/10%;

ListNode* plusOne(ListNode* head) {
Int carry = addoneutil(head);
if(carry>0) {
ListNode*newHead = new ListNode(carry);
newHead-> next = head;
Return newHead;}
Return head;
}
```

Ans :

```
#include
using namespace std;
struct ListNode {
    int val;
    ListNode*next;
    ListNode(int x) : val(x),next(nullptr) {}
};
```

```
int addOneutil(ListNode*node) {
    if(node==nullptr) return 1;
    int carry = addOneutil(node->next);
    int sum = node->val+carry;
    node->val = sum%10;
    return sum/10;
}
```

```
ListNode*plusOne(ListNode*head) {
```

```

int carry = addOneutil(head);
if(carry>0) {
    ListNode*newHead = new ListNode(carry);
    newHead->next=head;
    return newHead;
}
return head;

void printList(ListNode*head) {
    while(head) {
        cout<< head->val;
        if(head->next) cout << "->";
        head = head ->next;
    }
    cout<<endl;
}

ListNode*buildList(const vector& digits) {
    if(digits.empty()) return nullptr;
    ListNode*head = new ListNode(digits[0]);
    ListNode*current = head;
    for(int i = 1; i < digits.size(); ++i) {
        current ->next = new ListNode(digits[i]);
        current = current->next;
    }
    return head;
}

int main() {
    vector digits1 = {4,5,0};
    ListNode*num1 = buildList(digits1);
    cout<<"input";
    printList(num1);
    num1 = plusOne(num1);
    cout<<"output: ";
    printList(num1);
}
}

```

Updated one : -

```

#include<iostream>
#include<vector>
using namespace std;

namespace MyLinkedList {

```

```

struct ListNode {
    int val;
    ListNode* next;
    ListNode(int x) : val(x),next(nullptr) {}
};

int addOneutil(ListNode* node) {
    if(node==nullptr) return 1;
    int carry = addOneutil(node->next);
    int sum = node->val+carry;
    node->val = sum%10;
    return sum/10;
}

ListNode* plusOne(ListNode*head) {
    int carry = addOneutil(head);
    if(carry>0) {
        ListNode*newHead = new ListNode(carry);
        newHead->next=head;
        return newHead;
    }
    return head;
}

void printList(ListNode*head) {
    while(head) {
        cout<< head->val;
        if(head->next) cout << "->";
        head = head ->next;
    }
    cout<<endl;
}

ListNode*buildList(const vector<int>& digits) {
    if(digits.empty()) return nullptr;
    ListNode*head = new ListNode(digits[0]);
    ListNode*current = head;
    for(int i = 1; i < digits.size(); ++i) {
        current ->next = new ListNode(digits[i]);
        current = current->next;
    }
    return head;
}

int main() {
    vector<int> digits1 = {4,5,0};
    ListNode*num1 = buildList(digits1);
    cout<<"input";
    printList(num1);
    num1 = plusOne(num1);
}

```

```
    cout<<"output: ";  
    printList(num1);  
    return 0;  
}
```

Output :- Finished in 2 ms

input4->5->0

output: 4->5->1