List in C++ Standard Template Library(STL)

Assignment Solutions

PW SKILLS

Q1. Given a linked list. Each node contains the value that is either a or b. Your task is to insert a 'x' before every 'a' in the list.
The first line of input contains n, the number of elements in the linked list.
The second line of input contains the elements of the linked list.

INPUT
5
a b a b a

OUTPUT
x a b x a b x a

Solution:
Code Link: https://pastebin.com/g0JBBQad

test.cpp > main()

```cpp
#include <bits/stdc++.h>
#include <list>
using namespace std;

void add_x(list<char> &nums)
{
    for (auto it = nums.begin(); it != nums.end(); it++)
    {
        if (*it == 'a')
        {
            nums.insert(it, 'x');
        }
    }
    return;
}

int main()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
5
a b a b a
x a b x a b x a
```

Explanation:

Traverse the linked list. Whenever you find 'a', add a 'x' at that position. This will shift the remaining sequence by one place to the right.

Q2. Given a linked list, print the count of no. of 'x' you find in the list. The list contains only lowercase english letters.
The first line of input contains n, the number of elements in the linked list.
The second line of input contains the elements of the linked list.

INPUT
5
a x a b x

OUTPUT
2

Solution:
Code Link : https://pastebin.com/r3dJZ7Qc

test.cpp > main()

```cpp
#include <bits/stdc++.h>
#include <list>
using namespace std;

int count_x(list<char> &nums)
{
    int count = 0;
    for (auto it = nums.begin(); it != nums.end(); it++)
    {
        if (*it == 'x')
        {
            count++;
        }
    }
    return count;
}

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\Coding\cpp> cd "d:\Coding\cpp\" ; if ($?) { g++ test.cpp -o test } ; if ($?) { .\test }
5
a x b c x
2
```

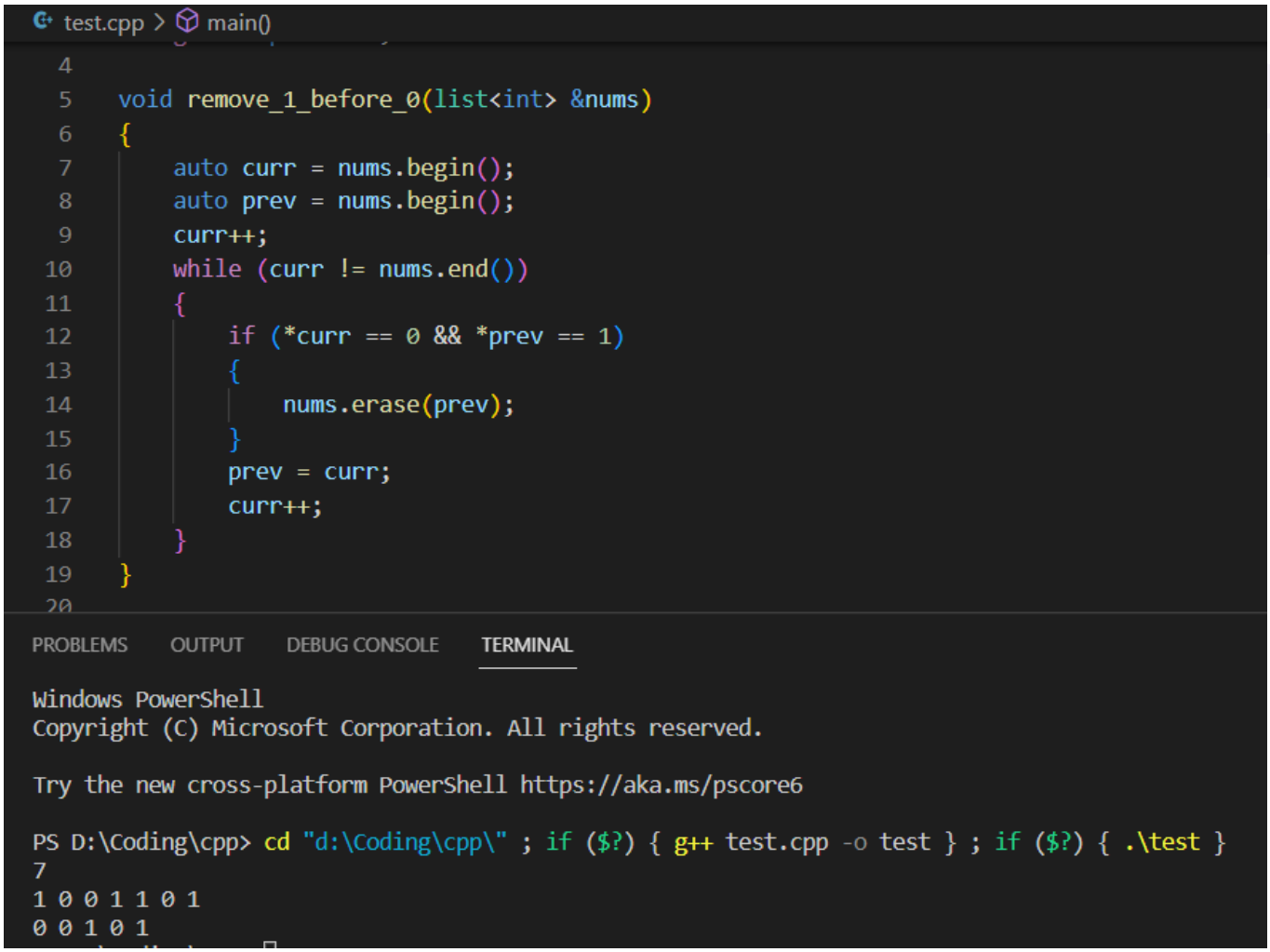
Explanation:
Traverse the linked list. Whenever you find 'x', increment the counter. Return the count at the end.

Q3. Given a linked list which stores only binary values i.e. 1's and 0's. Delete all the occurrences of 1 that appear before a 0.
The first line of input contains n, the number of elements in the linked list.
The second line of input contains the elements of the linked list.

INPUT
7
1 0 0 1 1 0 1

OUTPUT
0 0 1 0 1

Solution:
Code Link : https://pastebin.com/hFwsBgtp



Explanation:
Traverse the linked list. Whenever you find 1 at the prev position and 0 at the curr position, delete the prev node from the list. Now update the value of prev to curr and increment the value of curr.

Q4. Given a linked list, rotate the list by k positions to the left.
The first line of input contains n, the number of elements in the linked list, and k.
The second line of input contains the elements of the linked list.

Input
5 2
1 10 4 6 4

Output
4->6->4->1->10

Solution:
Code Link : https://pastebin.com/8GN49MNF

```
5   void rotate(list<int> &nums, int k)
6   {
7       k %= (int)nums.size();
8       while (k--)
9       {
10          int val = nums.front();
11          nums.pop_front();
12          nums.push_back(val);
13      }
14  }
15
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

5 2
1 10 4 6 4
4 6 4 1 10
PS D:\Coding\cpp>
```

Explanation:
To rotate the list just pop the element from the start of the list and insert it back at the end of the list. Do this k times to get the desired result.

Q5. Given 2 linked lists of equal length. For each element of the first list, print 1 if the corresponding element in the second list has the same value. Otherwise print 0.
The first line of input contains n, the size of the list.
The second line of input contains the elements of the first list.
The third line of input contains the elements of the second list.

INPUT
5 1 4 2 6 1
0 2 3 2 9 10

OUTPUT
0 0 1 0 1

Solution:

Code Link : https://pastebin.com/JNGG9tZW

```
5   vector<bool> areEqual(list<int> &nums1, list<int> &nums2)
6   {
7       auto it1 = nums1.begin(), it2 = nums2.begin();
8       vector<bool> ans;
9       while (it1 != nums1.end())
10      {
11          if (*it1 == *it2)
12          {
13              ans.push_back(1);
14          }
15          else
16          {
17              ans.push_back(0);
18          }
19          it1++;
20          it2++;
21      }
22      return ans;
23  }
24
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
5
1 4 2 6 10
2 3 2 9 10
0 0 1 0 1
```

Explanation:

In this we can use the 2 pointer approach. Both the pointers will move through the two lists simultaneously and give the corresponding result based on the comparison of the values in the 2 nodes.