C++ and Data Structures

Experiential Learning - Mini-project

Course Title: C++ and Data Structures
Course Code: 0701230310
Number of Credits: 02
Level: 02
Submitted to:Mr. Parag Narkhede, Assistant Professor,SIT
Submitted on: 27/10/2023

Submitted By	PRN
Garima Angelina	22070123042
Ishita Chaudhary	22070123048

AIM:

String Sorting

SOFTWARE USED:

Visual Studio Code

THEORY:

This code is written in C++ and demonstrates the implementation of a linked list data structure. It also includes the usage of a map container to count the occurrences of words in the linked list.

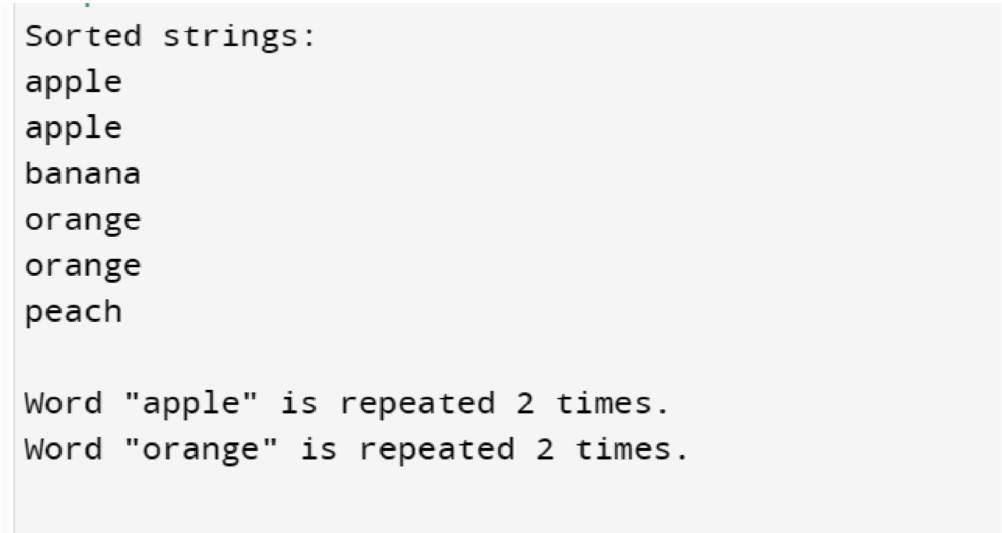
1. Linked List: A linked list is a data structure that consists of a sequence of nodes, where each node contains data and a reference to the next node in the sequence.
2. Map: A map is a container in C++ that stores elements in key-value pairs. It allows efficient retrieval of values based on their corresponding keys.

CODE:

```cpp
#include <iostream>
#include <string>
#include <map>
#include<algorithm>
using namespace std;
struct Node {
    string data;
    Node* next;
};
void insert(Node*& head, const string& data, map<string, int>& wordCount) {
    Node* newNode = new Node;
    newNode->data = data;
    newNode->next = head;
    head = newNode;
    wordCount[data]++;
}
void displayList(Node* head) {
    int i=0;
    string abc[100];
    Node* current = head;
    while (current != nullptr) {
        abc[i]=current->data;
        current = current->next;
        i++;
    }
    int size = i;
    sort(abc, abc + size);
    cout << "Sorted strings:"<<endl;
    for (int i = 0; i < size; ++i) {
        cout << abc[i] << endl;
    }
    cout << endl;
}
int main() {
    Node* head = nullptr;
    map<string, int> wordCount;
    insert(head, "apple", wordCount);
    insert(head, "banana", wordCount);
    insert(head, "peach", wordCount);
    insert(head, "orange", wordCount);
    insert(head, "apple", wordCount);
    insert(head, "orange", wordCount);
    displayList(head);
    for (const auto& pair : wordCount) {
        if (pair.second > 1) {
            cout << "Word \"" << pair.first << "\" is repeated " << pair.second << " times." << endl;
        }
    }

    return 0;
}
```

OUTPUT:



```
Sorted strings:
apple
apple
banana
orange
orange
peach

Word "apple" is repeated 2 times.
Word "orange" is repeated 2 times.
```

CONCLUSION:

A C++ program to sort a linked list of strings in alphabetical order and report any repeated word.