```
#include <iostream>
#include <string>
using namespace std;

struct Node {
    string data;
    Node* next;
};

class Queue {
private:
    Node* front;
    Node* rear;
public:
    Queue() : front(nullptr), rear(nullptr) {}

    void push(string item) {
        Node* newNode = new Node;
        newNode->data = item;
        newNode->next = nullptr;

        if (front == nullptr) {
            front = rear = newNode;
        } else {
            rear->next = newNode;
            rear = newNode;
        }
    }

    void pop() {
        if (front == nullptr) {
            cout << "The queue is empty" << endl;
            return;
        }

        Node* temp = front;
        front = front->next;
        delete temp;

        if (front == nullptr) {
            rear = nullptr;
        }
    }
```

```
    void printQueue() {
        Node* temp = front;
        while (temp != nullptr) {
            cout << temp->data << " ";
            temp = temp->next;
        }
        cout << endl;
    }

    bool isEmpty() {
        return front == nullptr;
    }

    int size() {
        int count = 0;
        Node* temp = front;
        while (temp != nullptr) {
            count++;
            temp = temp->next;
        }
        return count;
    }

    string frontElement() {
        if (front == nullptr) {
            cout << "The queue is empty" << endl;
            return "";
        }
        return front->data;
    }
};

int main() {
    Queue students;

    students.push("Alden");
    cout << "Inserting an item into an empty queue: " << endl;
    students.printQueue();
    cout << "\n";

    students.push("Ben");
    students.push("Charles");
    cout << "Inserting an item into a non-empty queue: " << endl;
    students.printQueue();
```

```
    cout << "\n";

    students.pop();
    cout << "Deleting an item from a queue of more than one item: " << endl;
    students.printQueue();
    cout << "\n";

    students.pop();
    students.pop();
    cout << "Deleting an item from a queue with one item: " << endl;
    students.printQueue();
    cout << NULL;
    return 0;
}
```