```cpp

//Sartaj Khan
//Date: 26/04/23

#include "LZWTable.h"
#include <cassert>

LZWTable::LZWTable(uint16_t aInitialCharacters) :
    fIndex(0),
    fInitialCharacters(aInitialCharacters)
{}

void LZWTable::initialize() {
    for (size_t i = 0; i < 128; i++) {
        fEntries[fIndex] = PrefixString((char)i);
        fEntries[fIndex++].setCode(i);
    }
}

const PrefixString& LZWTable::lookupStart(char aK) const noexcept {
    assert(0 <= aK < fIndex);
    PrefixString found = PrefixString();
    for (PrefixString aPrefixString : fEntries) {
        if (aPrefixString.K() == aK) {
            found = aPrefixString;
            break;
        }
    }
    return found;
}

bool LZWTable::contains(PrefixString& aWK) const noexcept {
    assert(aWK.w() != static_cast<uint16_t>(-1));
    for (size_t index = fIndex - 1; index >= aWK.w(); index--) {
        if (fEntries[index] == aWK) {
            aWK = fEntries[index];
            return true;
        }
    }
    return false;
}

void LZWTable::add(PrefixString& aWK) noexcept {
    assert(aWK.w() != static_cast<uint16_t>(-1));
    aWK.setCode(fIndex);
    fEntries[fIndex++] = aWK;
}
```