

```

//
// KeyProvider.cpp
// midSem
//
// Created by H M Asfaq Ahmed Shihab on 25/4/2024.
//
#include "KeyProvider.hpp"
#include <cctype>
#include <cassert>

std::string KeyProvider::preprocessString(const std::string& aString)
    noexcept {
    std::string result;
    for (char ch : aString) {
        if (std::isalpha(ch)) {
            result += std::toupper(ch);
        }
    }
    return result;
}

KeyProvider::KeyProvider(const std::string& aKeyword, const
    std::string& aSource) noexcept {
    std::string processedKeyword = preprocessString(aKeyword);
    std::string processedSource = preprocessString(aSource);

    while (fKeys.length() < processedSource.length()) {
        fKeys += processedKeyword;
    }

    fKeys = fKeys.substr(0, processedSource.length());
    fIndex = 0;
}

char KeyProvider::operator*() const noexcept {
    return fKeys[fIndex];
}

KeyProvider& KeyProvider::operator++() noexcept {
    if (fIndex < fKeys.size()) fIndex++;
    return *this;
}

KeyProvider KeyProvider::operator++(int) noexcept {
    KeyProvider temp = *this;
    ++(*this);
    return temp;
}

bool KeyProvider::operator==(const KeyProvider& aOther) const
    noexcept {

```

```

        return fIndex == aOther.fIndex && fKeys == aOther.fKeys;
    }

    bool KeyProvider::operator!=(const KeyProvider& aOther) const
        noexcept {
        return !(*this == aOther);
    }

    KeyProvider KeyProvider::begin() const noexcept {
        KeyProvider temp(*this);
        temp.fIndex = 0;
        return temp;
    }

    KeyProvider KeyProvider::end() const noexcept {
        KeyProvider temp(*this);
        temp.fIndex = fKeys.size();
        return temp;
    }
}

```