作业 1:

main.cpp

```
#include <iostream>
#include "sq_list.h"
using namespace std;

void PrintElem(const char &e) {
    cout << e << " ";
}

int main() {
    // 创建顺序表 A 和 B，类型为 char
    SqList<char> A, B;

    // 向顺序表 A 插入元素
    A.Insert(1, 'A');
    A.Insert(2, 'B');
    A.Insert(3, 'C');
    A.Insert(4, 'D');

    // 向顺序表 B 插入元素
    B.Insert(1, 'B');
    B.Insert(2, 'C');
    B.Insert(3, 'E');
    B.Insert(4, 'F');
    B.Insert(5, 'G');
```

```
cout << "线性表 A 的元素为: ";
A.Traverse(PrintElem);
cout << endl;

cout << "线性表 B 的元素为: ";
B.Traverse(PrintElem);
cout << endl;

// ------------------ 并集操作 ------------------
// 并集：先将 A 的所有元素加入结果，再将 B 中不在 A 中的元素加入结果
SqList<char> UnionAB = A; // 先复制 A
for (int i = 1; i <= B.Length(); i++) {
    char e;
    B.GetElem(i, e); // 取 B 的第 i 个元素
    bool found = false;
    // 检查 A 中是否有该元素
    for (int j = 1; j <= A.Length(); j++) {
        char a;
        A.GetElem(j, a);
        if (a == e) {
            found = true;
            break;
        }
    }
    // 如果 A 中没有，则加入并集
    if (!found) UnionAB.Insert(UnionAB.Length() + 1, e);
}
```

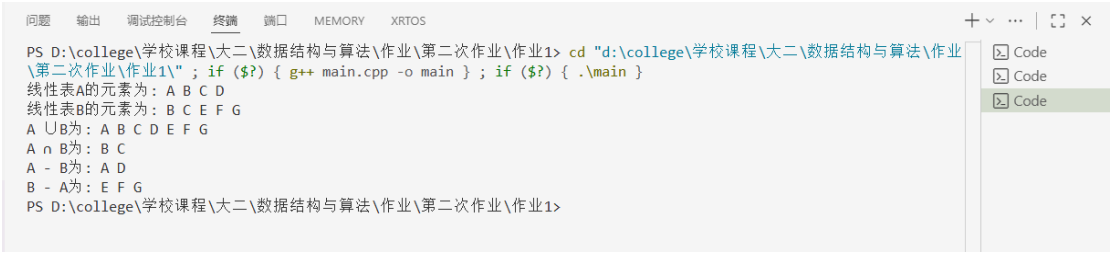
```
// ------------------ 交集操作 ------------------
// 交集：A 和 B 都包含的元素
SqList<char> InterAB;
for (int i = 1; i <= A.Length(); i++) {
    char e;
    A.GetElem(i, e); // 取 A 的第 i 个元素
    for (int j = 1; j <= B.Length(); j++) {
        char b;
        B.GetElem(j, b);
        if (e == b) { // 如果 B 中也有该元素
            InterAB.Insert(InterAB.Length() + 1, e); // 加入交集
            break; // 找到后跳出内层循环
        }
    }
}

// ------------------ 差集 A-B 操作 ------------------
// 差集 A-B：A 中有但 B 中没有的元素
SqList<char> DiffAB;
for (int i = 1; i <= A.Length(); i++) {
    char e;
    A.GetElem(i, e);
    bool found = false;
    for (int j = 1; j <= B.Length(); j++) {
        char b;
        B.GetElem(j, b);
```

```
            if (e == b) {
                found = true;
                break;
            }
        }
        // 如果 B 中没有该元素，则加入差集
        if (!found) DiffAB.Insert(DiffAB.Length() + 1, e);
}

// ------------------ 差集 B-A 操作 ------------------
// 差集 B-A：B 中有但 A 中没有的元素
SqList<char> DiffBA;
for (int i = 1; i <= B.Length(); i++) {
        char e;
        B.GetElem(i, e);
        bool found = false;
        for (int j = 1; j <= A.Length(); j++) {
            char a;
            A.GetElem(j, a);
            if (e == a) {
                found = true;
                break;
            }
        }
        // 如果 A 中没有该元素，则加入差集
        if (!found) DiffBA.Insert(DiffBA.Length() + 1, e);
}
```

```
    // ------------------ 输出结果 ------------------
    cout << "A ∪ B 为: ";
    UnionAB.Traverse(PrintElem); // 并集输出
    cout << endl;

    cout << "A ∩ B 为: ";
    InterAB.Traverse(PrintElem); // 交集输出
    cout << endl;

    cout << "A - B 为: ";
    DiffAB.Traverse(PrintElem); // 差集 A-B 输出
    cout << endl;

    cout << "B - A 为: ";
    DiffBA.Traverse(PrintElem); // 差集 B-A 输出
    cout << endl;

    // 程序结束
}
```

还调用了 sq_list.h 头文件

运行结果：



作业 2:

main.cpp

```
#include <iostream>
#include "simple_lk_list.h"                    // 简单线性链表类模板
using namespace std;

void PrintElem(const int &e) {
    cout << e << " ";
}

SimpleLinkList<int> DifferenceSet(const SimpleLinkList<int>& la, const SimpleLinkList<int>& lb);

int main(){
    // 创建两个递增有序集合 A 和 B
    SimpleLinkList<int> la, lb;
    // 示例插入数据
    la.Insert(1, 1);
    la.Insert(2, 3);
    la.Insert(3, 5);
    la.Insert(4, 7);
    lb.Insert(1, 3);
```

```
    lb.Insert(2, 4);
    lb.Insert(3, 7);

    cout<< "集合 A 为" << endl;
    la.Traverse(PrintElem);
    cout << endl;

    cout<< "集合 B 为" << endl;
    lb.Traverse(PrintElem);
    cout << endl;

    // 求差集 C = A - B
    SimpleLinkList<int> lc = DifferenceSet(la, lb);
    cout << "A - B 的差集为: ";
    lc.Traverse(PrintElem);
    cout << endl;
    return 0;
}

SimpleLinkList<int> DifferenceSet(const SimpleLinkList<int>& la, const
SimpleLinkList<int>& lb) {
    SimpleLinkList<int> lc;
    int i = 1, j = 1;
    int a, b;
    int lenA = la.Length(), lenB = lb.Length();
    while (i <= lenA && j <= lenB) {
        la.GetElem(i, a);
```

```
            lb.GetElem(j, b);
            if (a < b) {
                lc.Insert(lc.Length() + 1, a);
                i++;
            }
            else if (a == b) {
                i++;
                j++;
            }
            else {
                j++;
            }
        }
        //A 剩下的都属于差集
        while (i <= lenA) {
            la.GetElem(i, a);
            lc.Insert(lc.Length() + 1, a);
            i++;
        }
        return lc;
}
```

还调用了 simple_lk_list.h 和 node.h 头文件

运行结果：


问题
输出
调试控制台
终端
端口
MEMORY
XRTOS
powershell
Code
PS D:\college\学校课程\大二\数据结构与算法\作业\第二次作业\作业2> cd "d:\college\学校课程\大二\数据结构与算法\作业\第二次作业\作业2\" ; if ($?) { g++ main.cpp -o main } ; if ($?) { .\main }
集合A为
1 3 5 7
集合B为
3 4 7
A - B 的差集为：1 5
PS D:\college\学校课程\大二\数据结构与算法\作业\第二次作业\作业2>