**南 京 师 范 大 学**

**《数据结构》**

**课程设计报告**



**题 目： 压缩软件**

**学 院：计算机与电子信息学院/人工智能学院**

**专 业： 计算机科学与技术**

**姓 名： 刘浩天**

**学 号： 19220211**

**任课教师： 吉根林**

**二O二** **四 年 二 月 二十八 日**

一、多种排序算法

（一）希尔排序

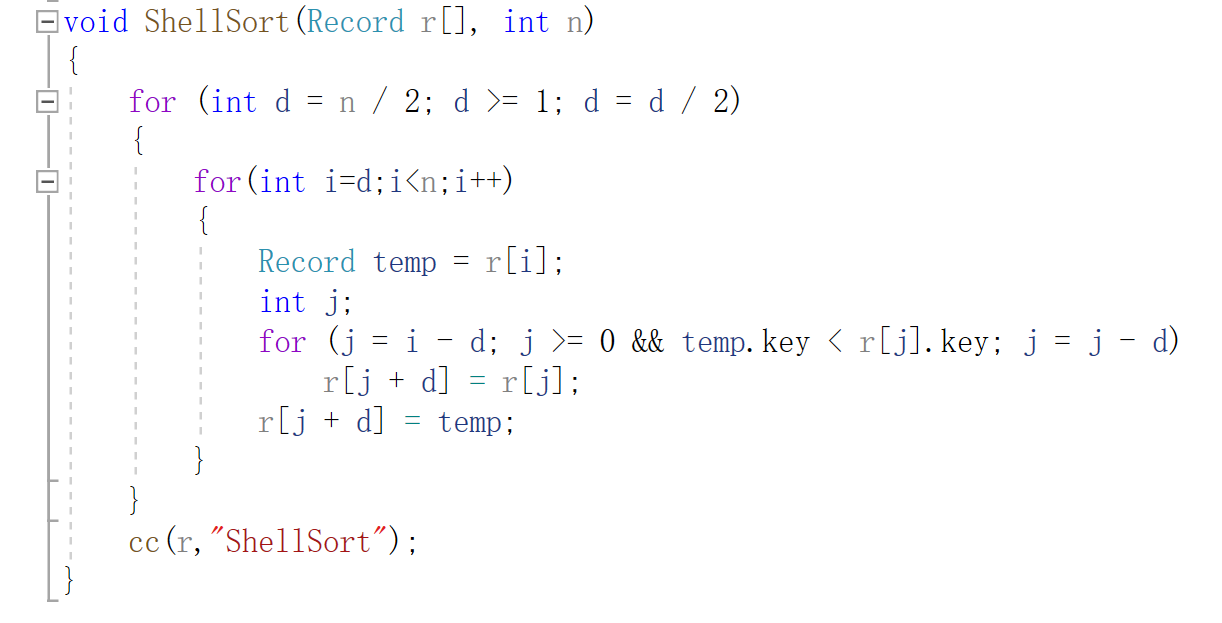
1、希尔排序算法原理

先将待排序列划分为若干小序列，在这些小序列中进行插入排序；

然后逐步扩大小序列的长度，减少小序列的个数，这样使待排序列逐渐处于更有序的状态；

最后对全体序列进行一次直接插入排序，从而完成排序。

2、希尔排序源代码

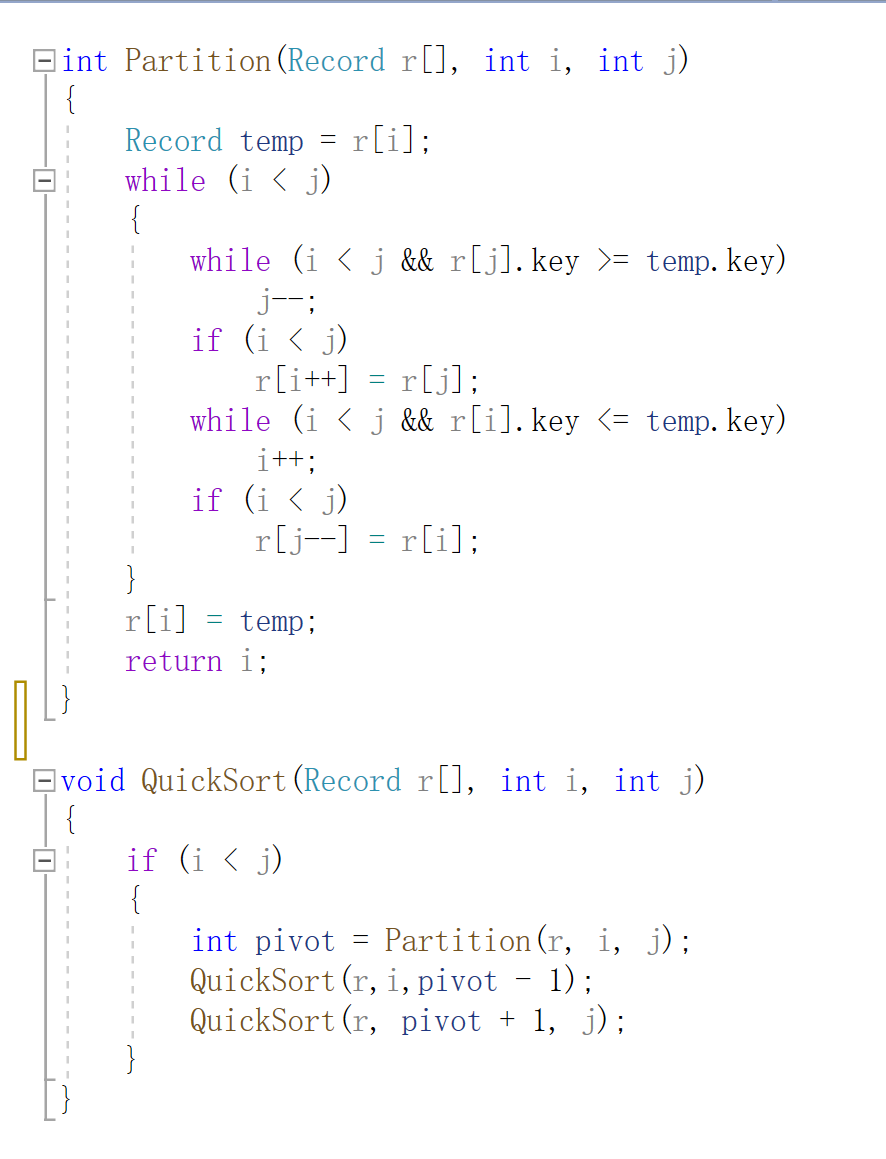


（二）快速排序

1、快速排序算法原理

首先选一个轴值（即比较的基准），通过一趟排序将待排序记录分割成独立的两部分，前一部分记录的关键码均小于或等于轴值，后一部分记录的关键码均大于或等于轴值，然后分别对这两部分重复上述方法，直到整个序列有序。

2、快速排序源代码



（三）堆排序

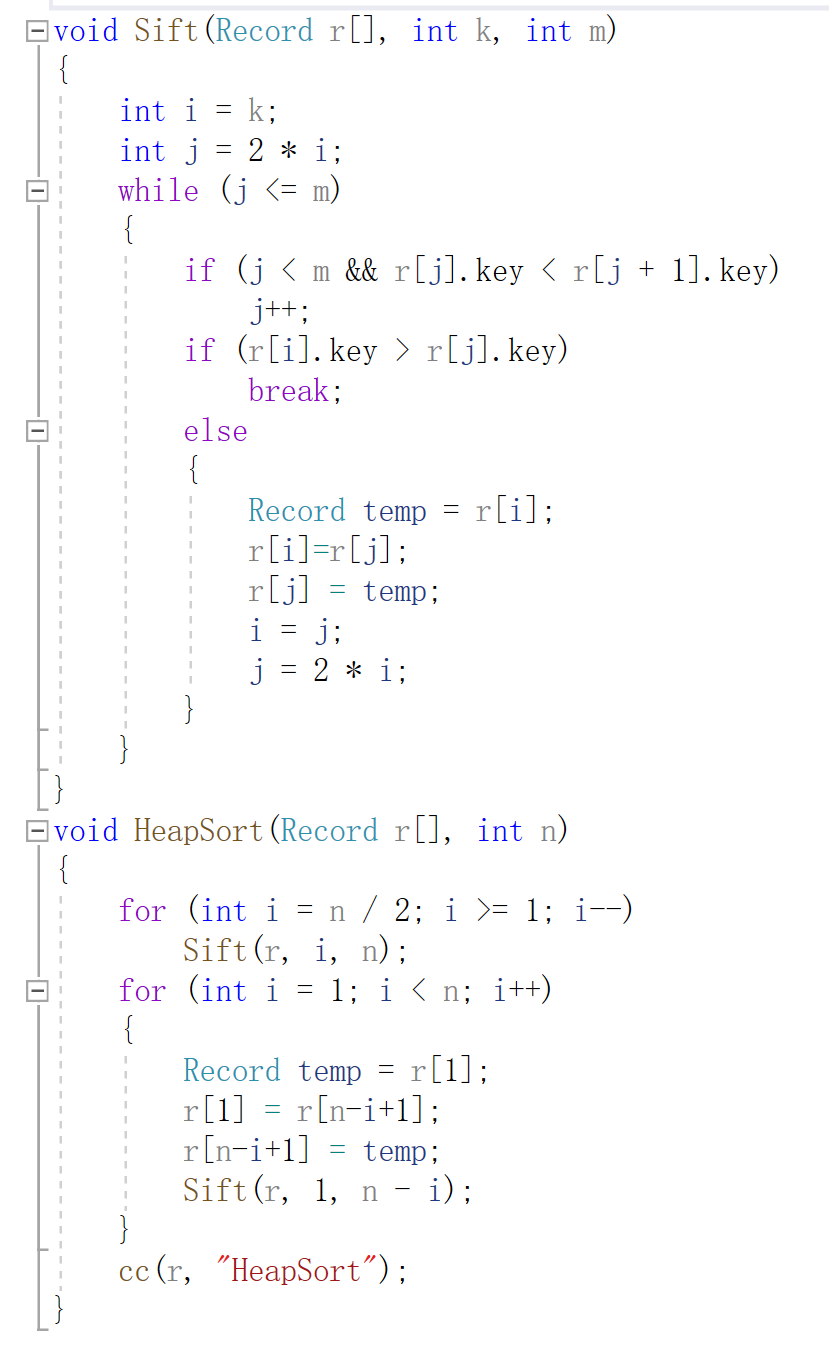
1、堆排序算法原理

首先将待排序的记录序列构造成一个堆，此时，选出了堆中所有记录的最大者，然后将它从堆中移走，并将剩余的记录再调整成堆，这样又找出了次小的记录，以此类推，直到堆中只有一个记录。

如何用无序序列建成一个堆：以n/2位置的树为根，调整以其为根的所有子树。从根节点左右孩子中挑一个大的，然后根节点与大孩子相比，若小于大孩子则交换，若大于或等于，则break，交换完后，以被交换孩子为根节点，重复上述操作。

初始堆建成后，则选出了最大值（在堆顶，data[1]位置），将堆顶移走，即将data[1]与data[最后位置]交换，再重新建堆，注意数据个数减少1个。重复上述操作。

2、堆排序源代码



（四）归并排序

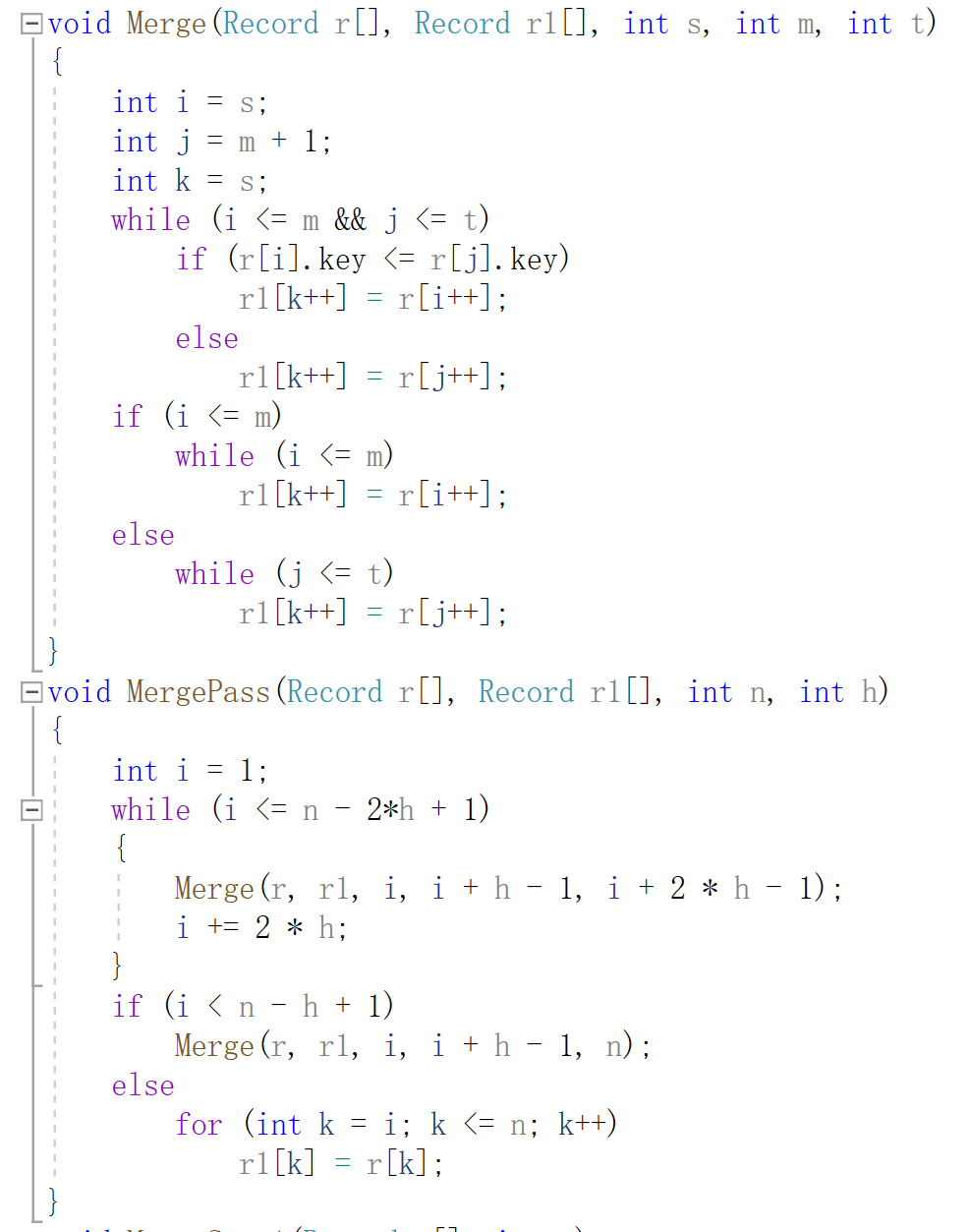
1、归并排序算法原理

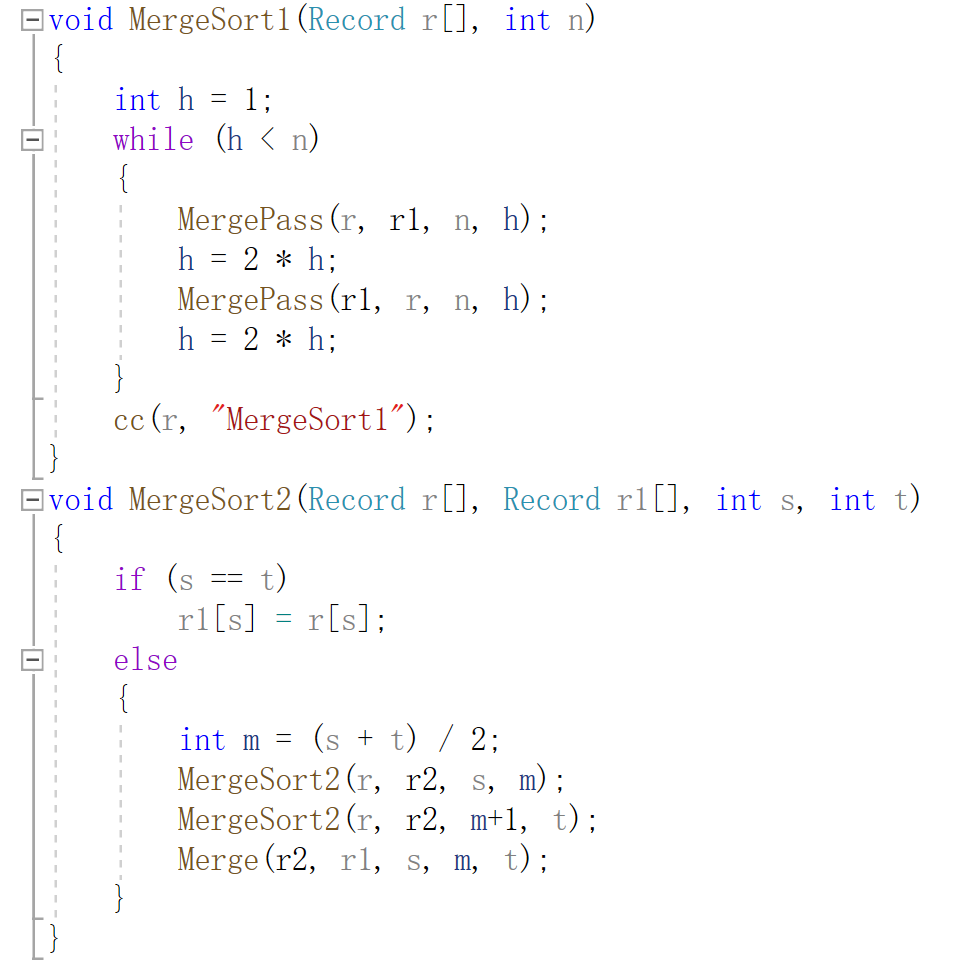
将一个具有n个待排序记录的序列看成是n个长度为1的有序序列，然后进行两两归并，得到n/2个长度为2的有序序列，再进行两两归并，得到n/4个长度为4的有序序列，……，直至得到一个长度为n的有序序列为止。

完成一趟归并，结束时需要分3种情况，（1）刚好剩下两个子数组，且长度都为h（2）剩下两个数组，其中一个数组长度小于h（3）只剩一个数组直接接到r1数组后面

控制二路归并的结束：开始时，有序序列的长度h=1，结束时，有序序列的长度h=n，用有序序列的长度来控制排序的结束。

2、归并排序源代码





|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 排序方法 | 时间复杂度（平均） | 时间复杂度  （最好） | 时间复杂度  （最坏） | 空间复杂度 | 稳定性 |
| 希尔排序 |  |  |  |  | 不稳定 |
| 快速排序 |  |  |  |  | 不稳定 |
| 堆排序 |  |  |  |  | 不稳定 |
| 归并排序 |  |  |  |  | 稳定 |

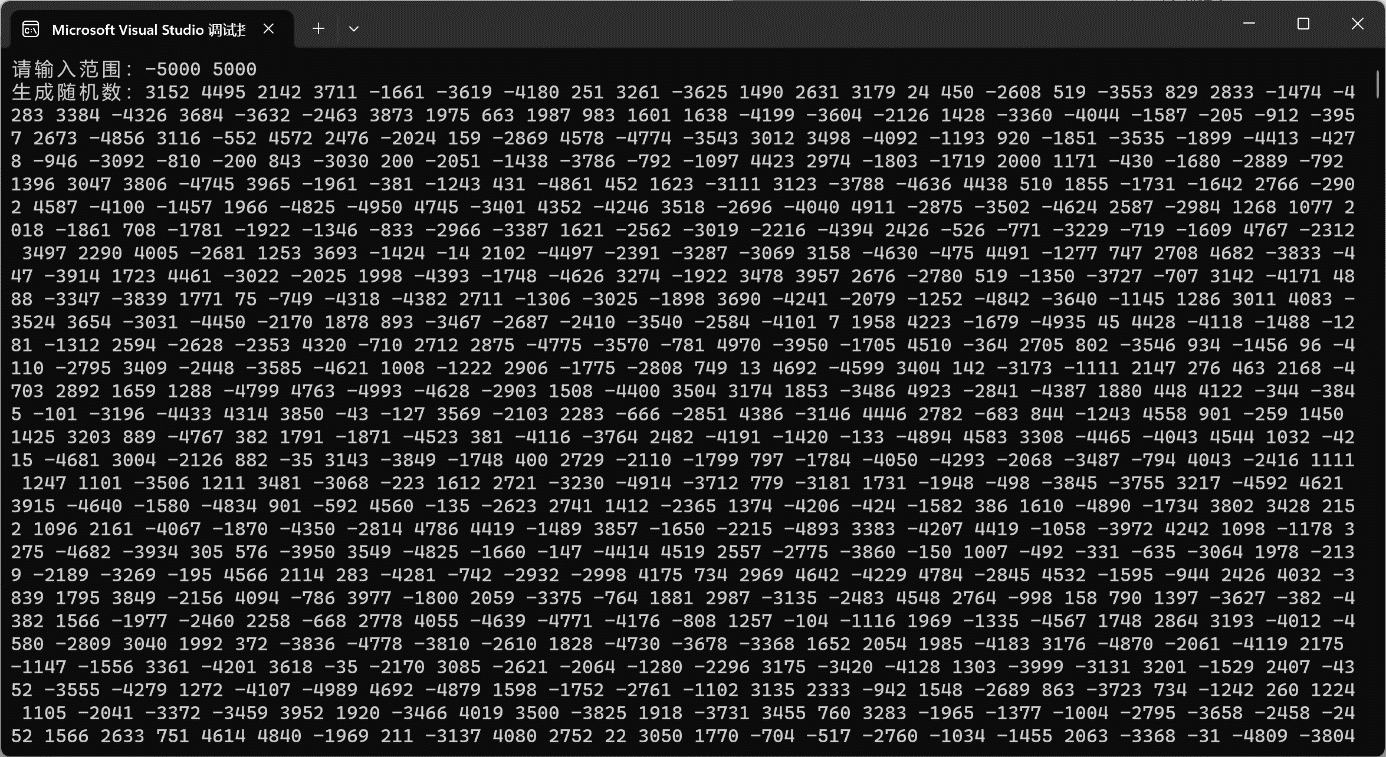
（五）四种排序算法比较

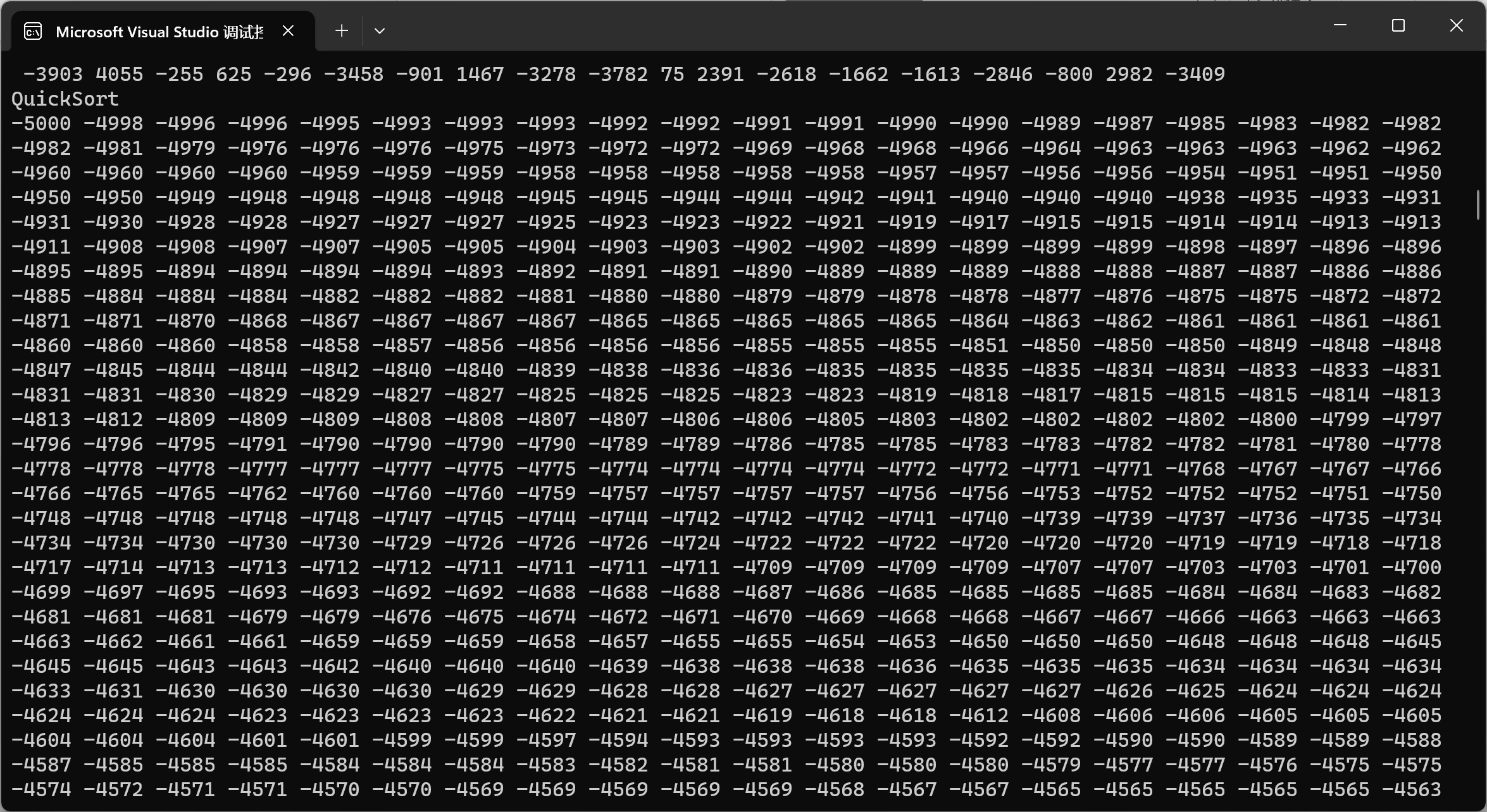
（六）程序设计

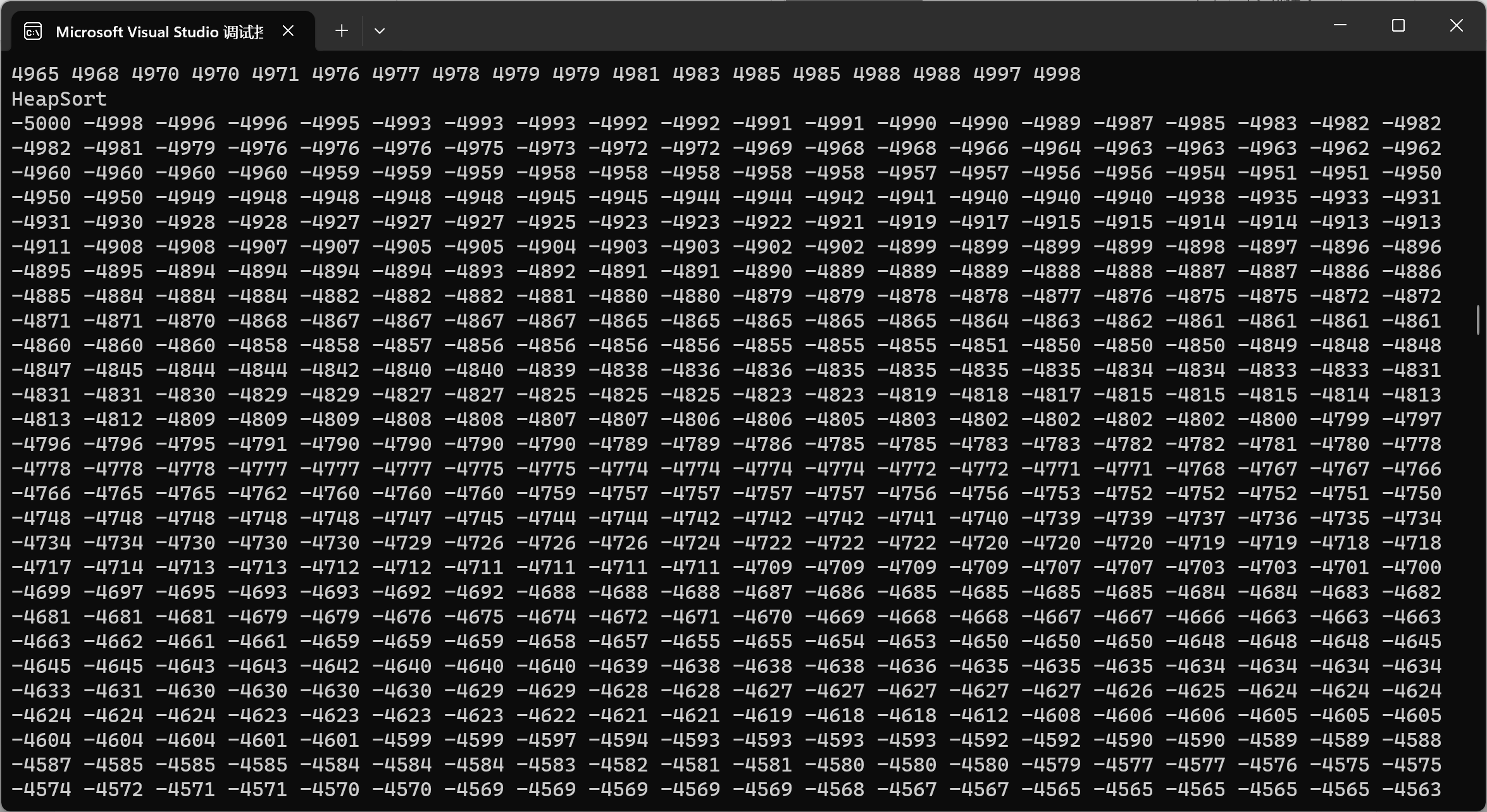
1、程序设计要求

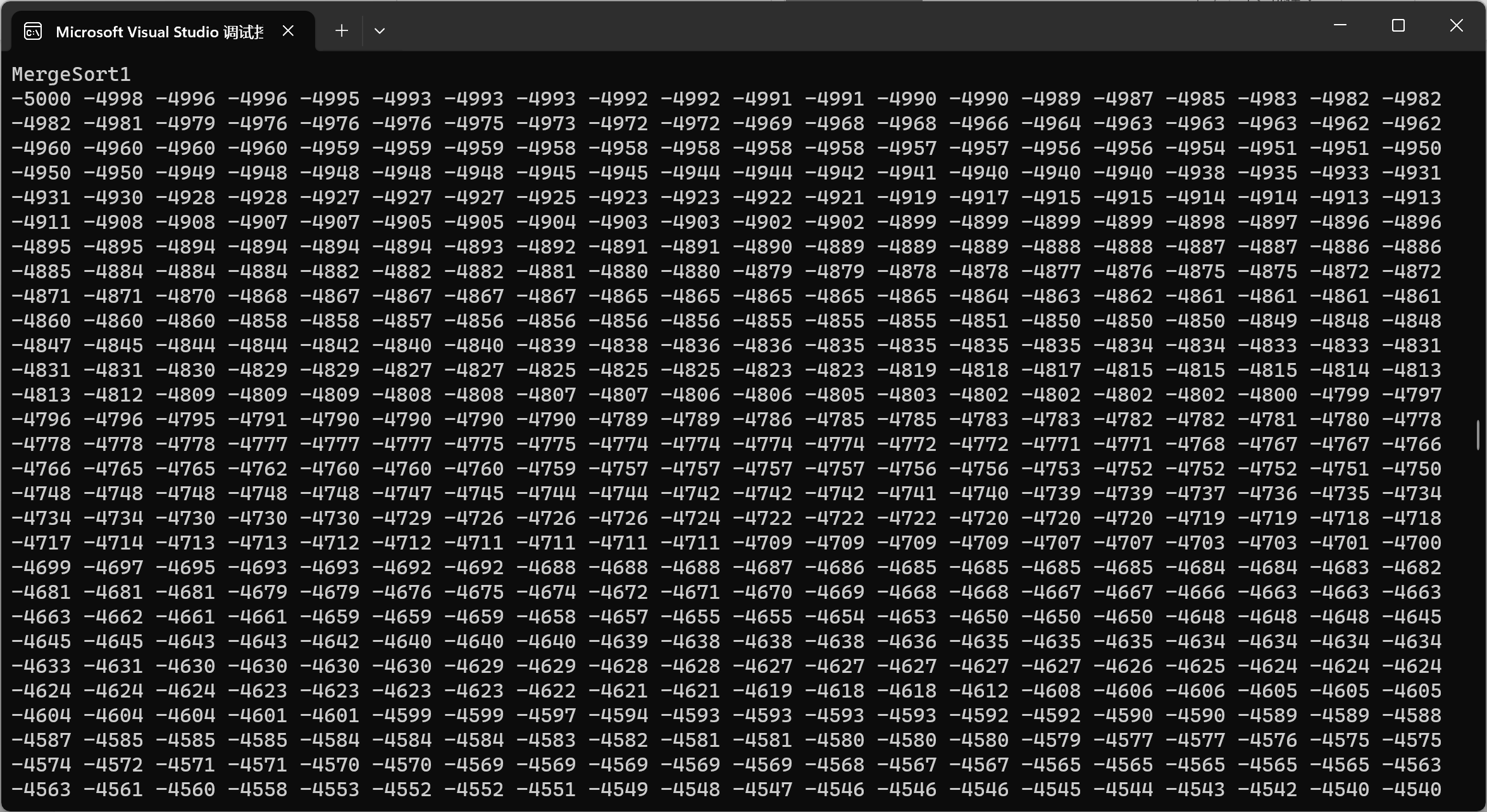
编程实现希尔、快速、堆排序、归并排序算法。要求首先随机产生10000个数据存入磁盘文件，然后读入数据文件，分别采用不同的排序方法进行排序并将结果存入文件中。

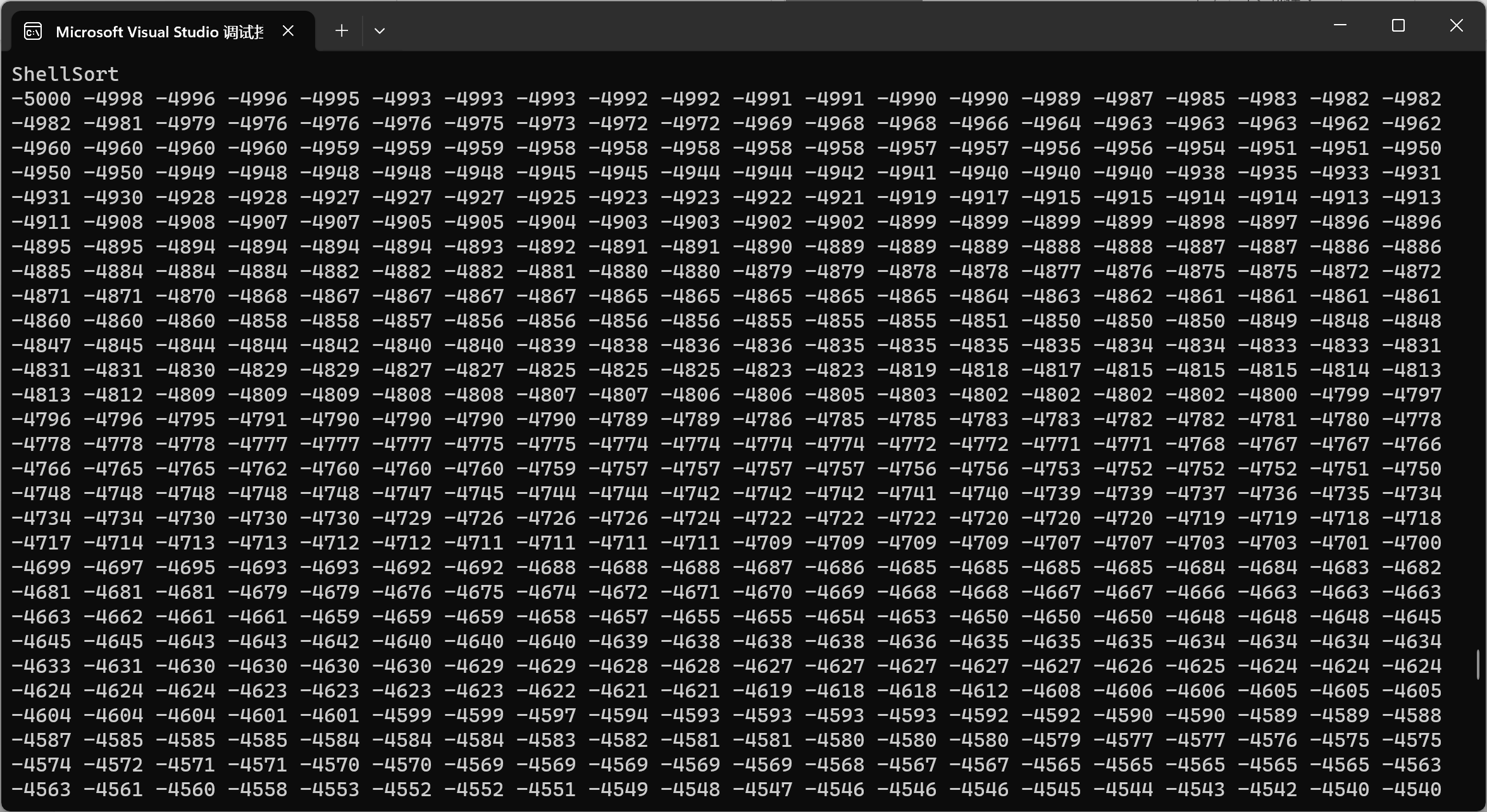
2、程序测试案例

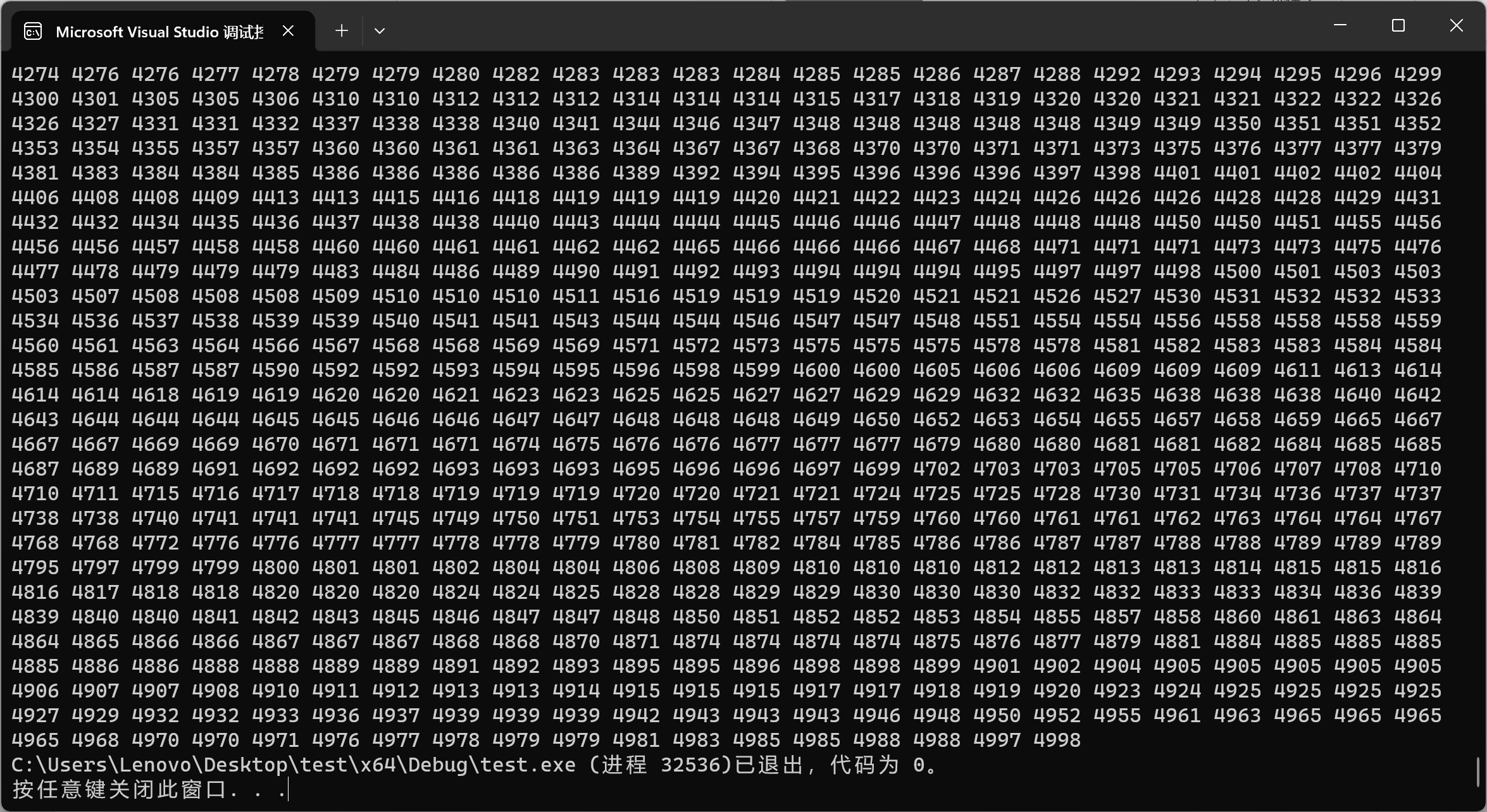












二、压缩软件

（一）程序设计要求

建立一个文本文件A（可以是C/C++源程序），统计该文件中各字符频率。首先对各字符进行Huffman编码，将该文件翻译成Huffman编码文件B；然后将Huffman编码文件译码成文件C，并对文件A与C进行比较。

（二）各模块要实现的功能

设计HuffmanTree类等类模块以及调试模块。

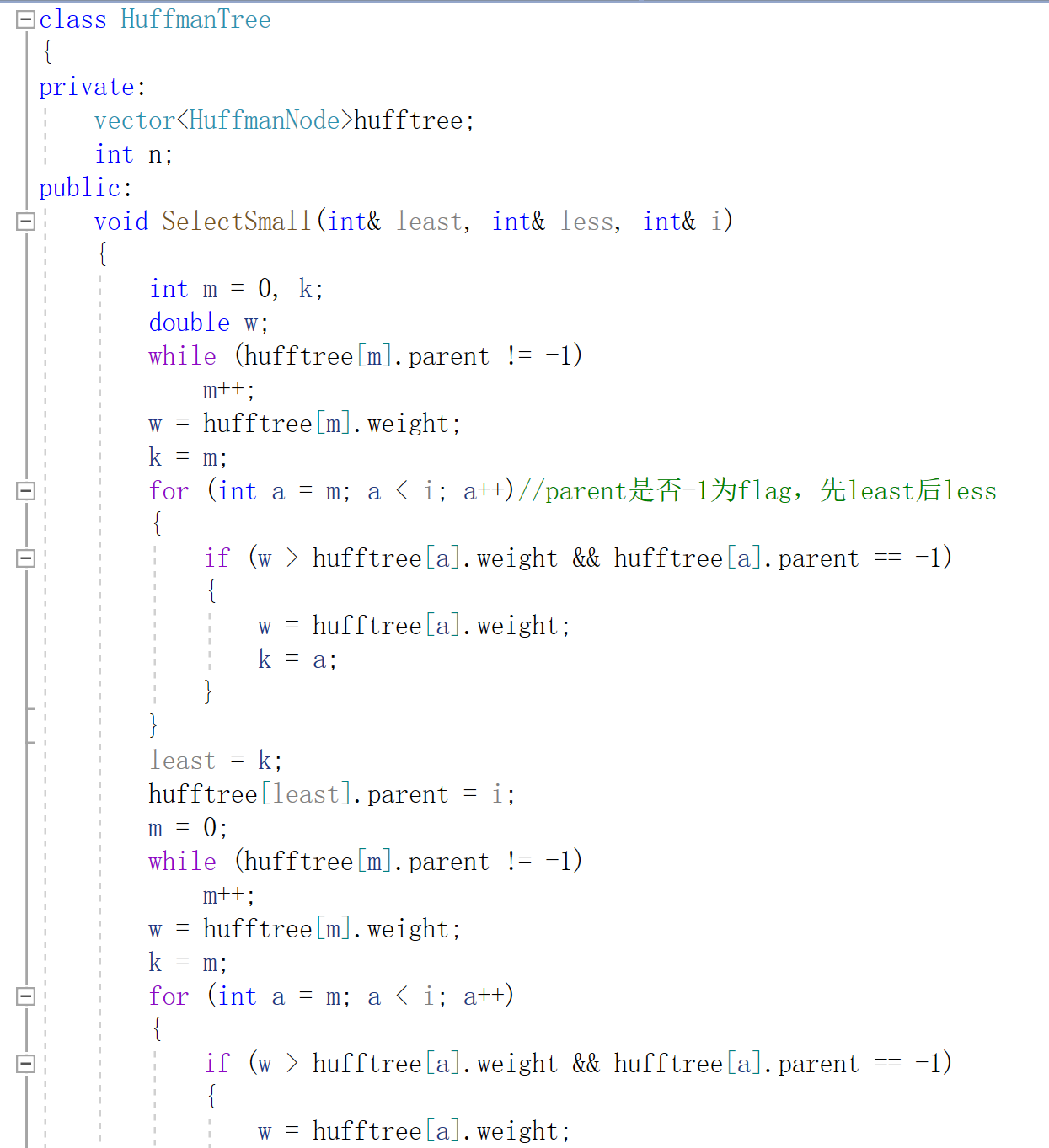
1. HuffmanTree类包含HuffmanNode型结点和结点数量，并且有相应的构造、编码和译码等函数。
2. HuffmanNode型结点包含字符数据、字符频率以及结点的父亲结点和孩子结点。

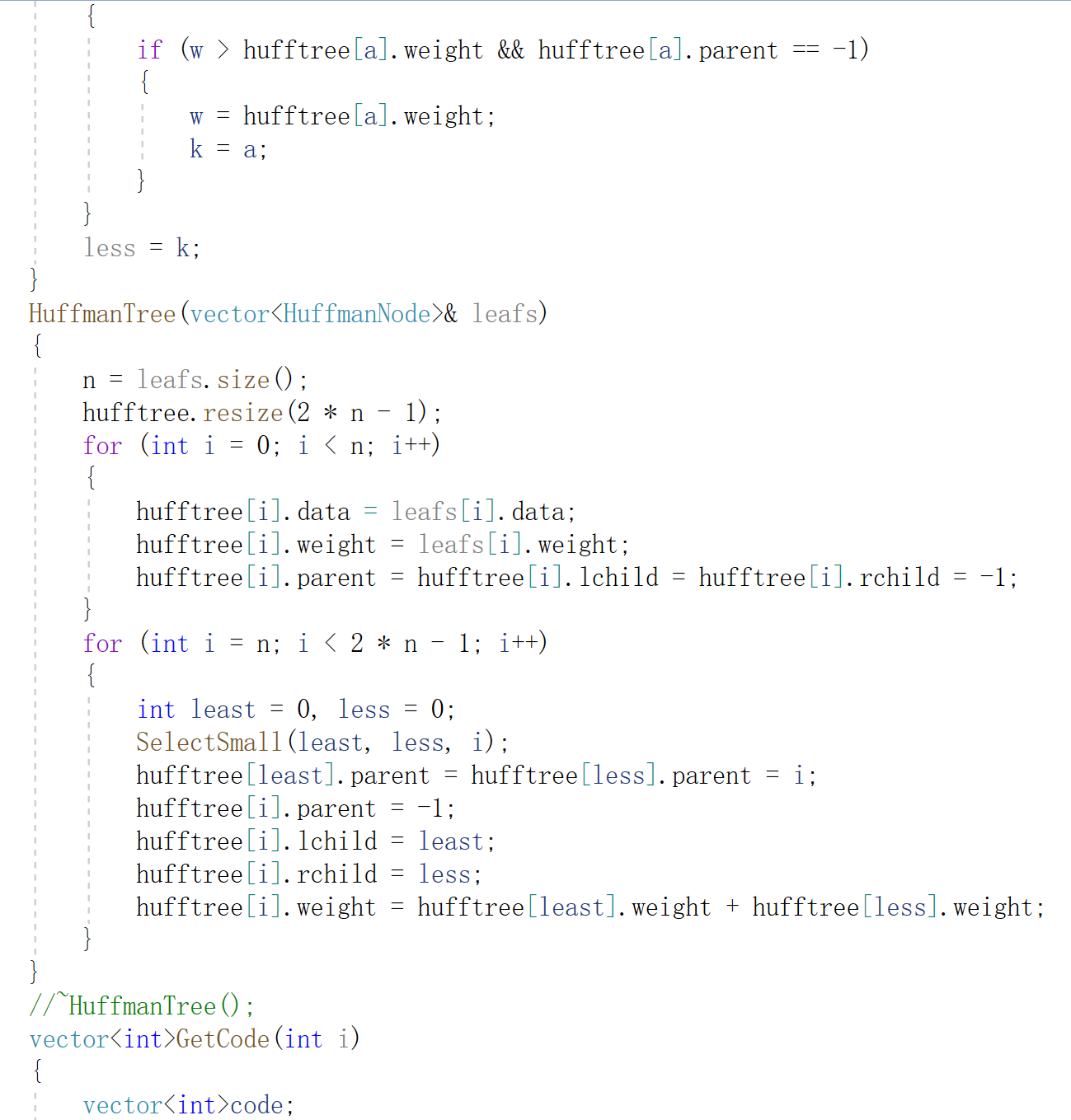
（三）程序结构

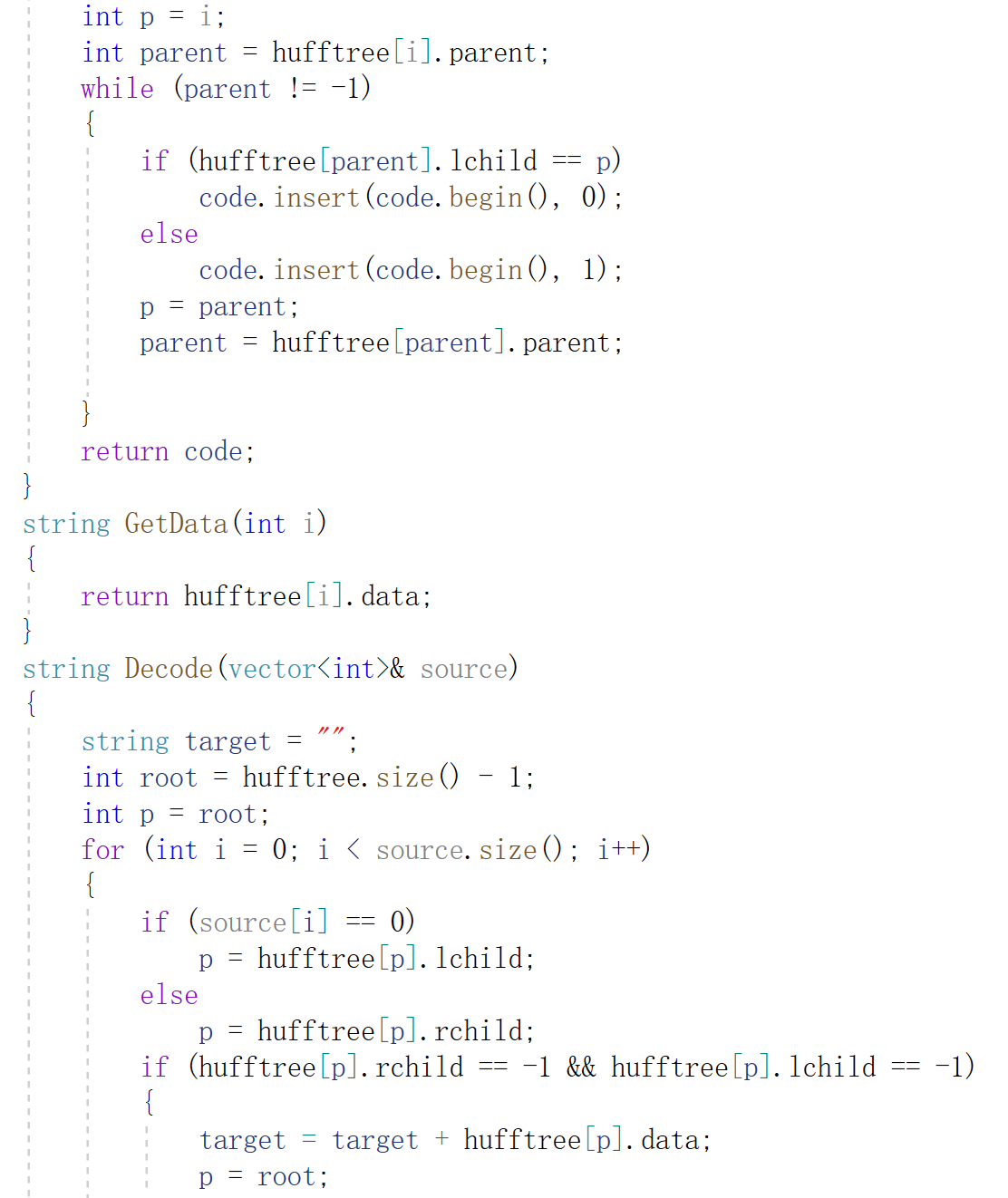
表3.1 HuffmanTree类

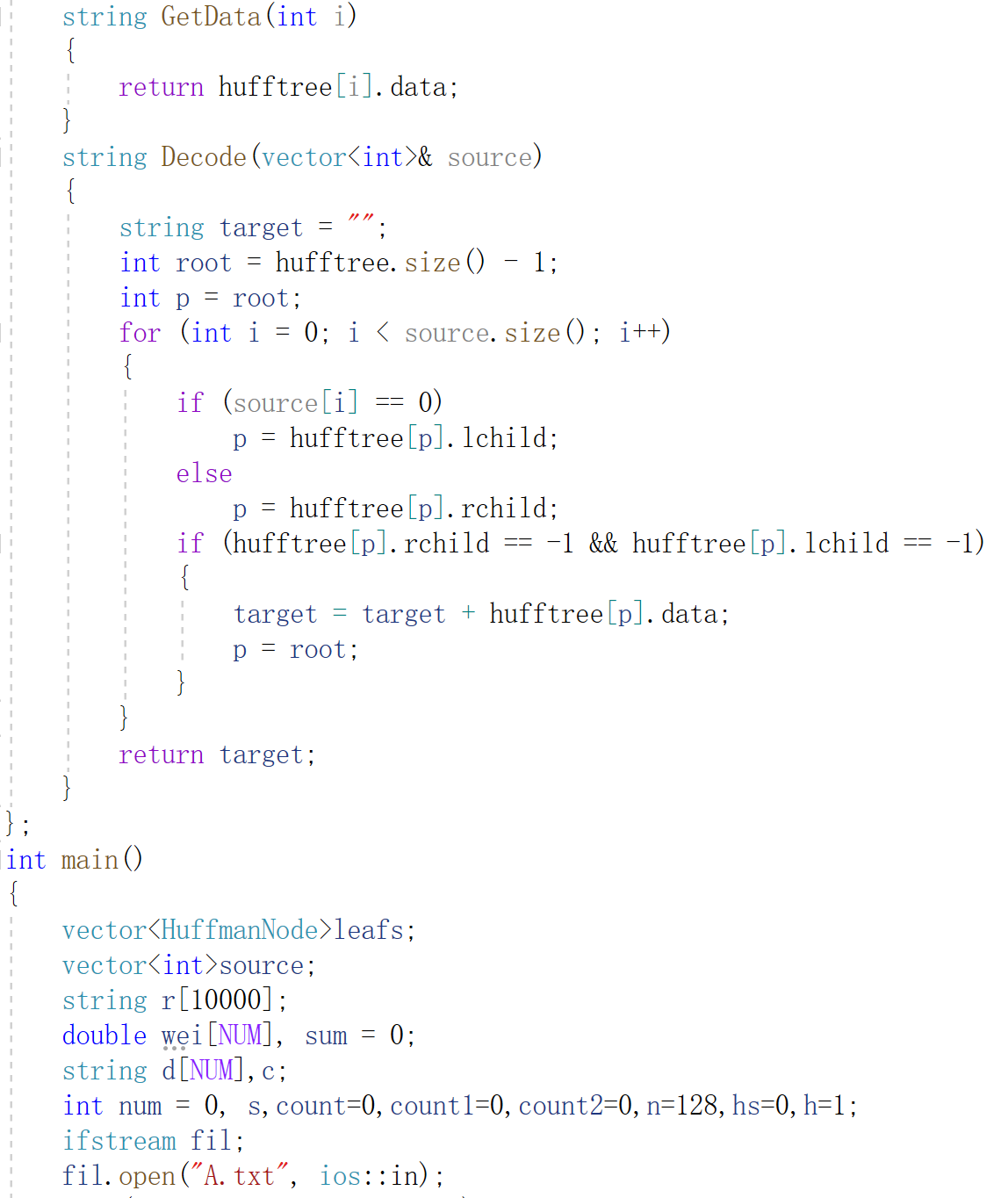
|  |  |
| --- | --- |
| 成员 | 作用 |
| vector<HuffmanNode>hufftree; | 记录结点信息 |
| int n; | 记录结点数量 |
| void SelectSmall(int& least, int& less, int& i); | 选出待选结点中频率最小的两个结点 |
| HuffmanTree(vector<HuffmanNode>& leafs); | 构造Huffman树 |
| vector<int>GetCode(int i)； | 编码 |
| string GetData(int i)； | 返回字符数据 |
| string Decode(vector<int>& source)； | 译码 |

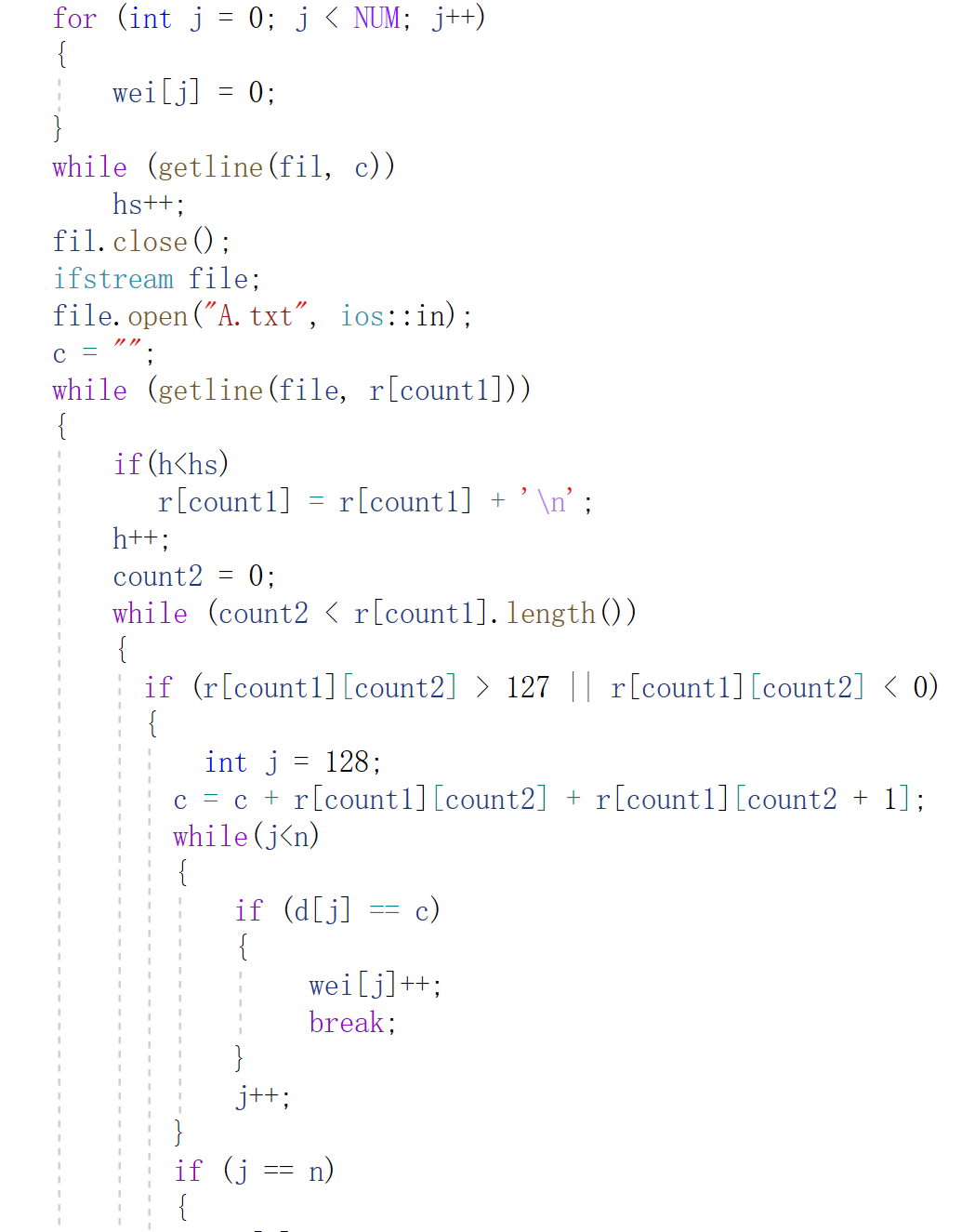
相关源代码

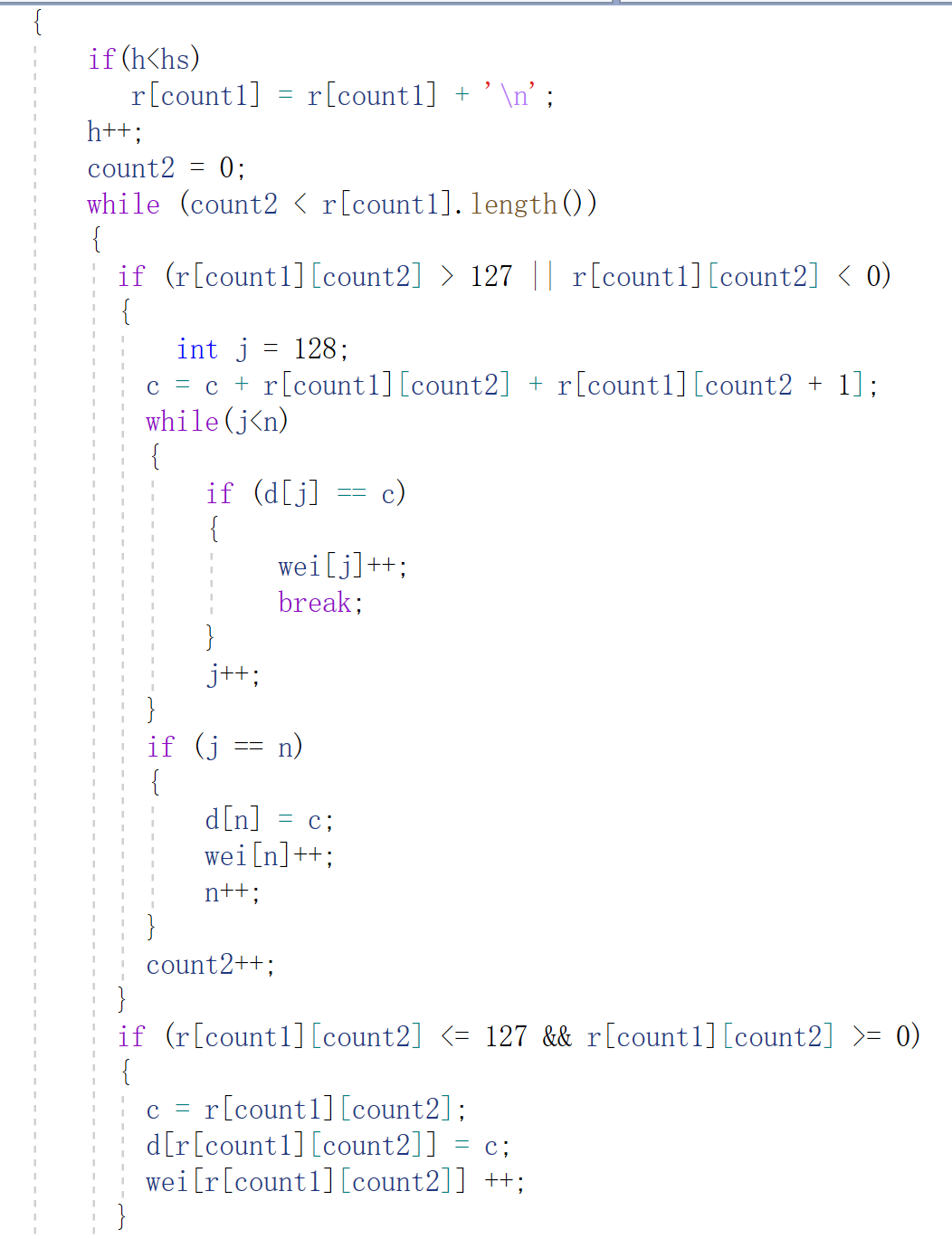


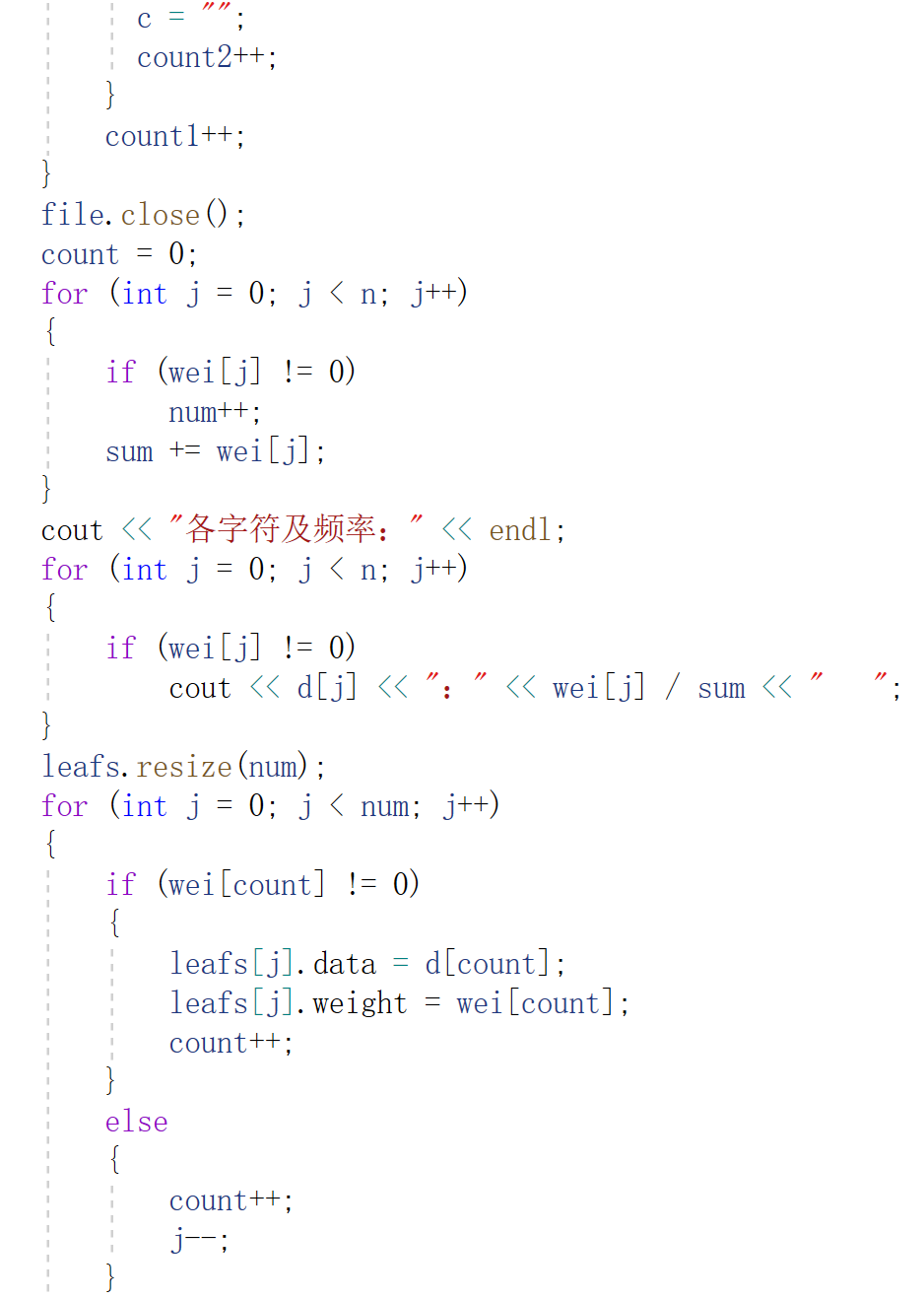


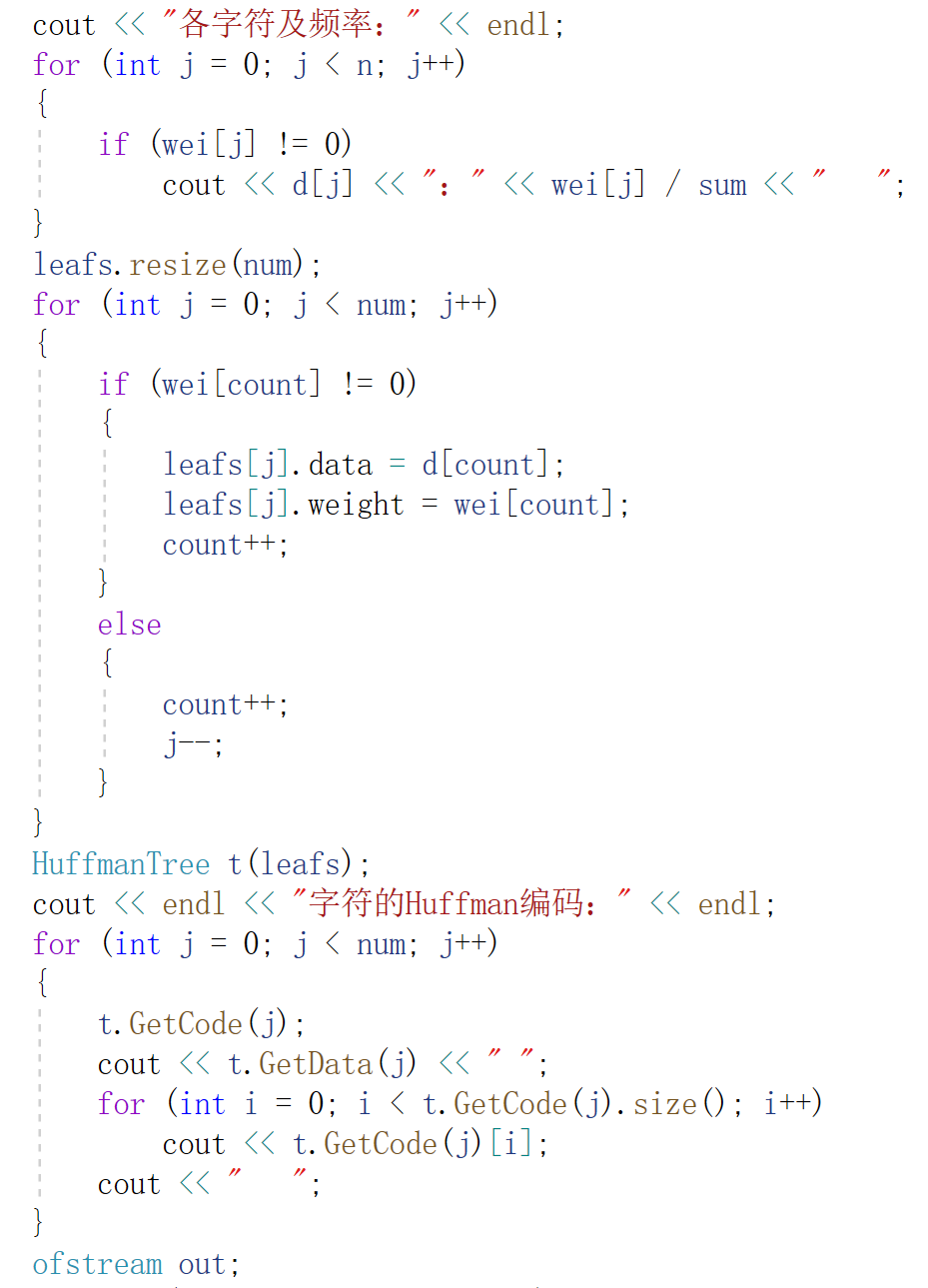


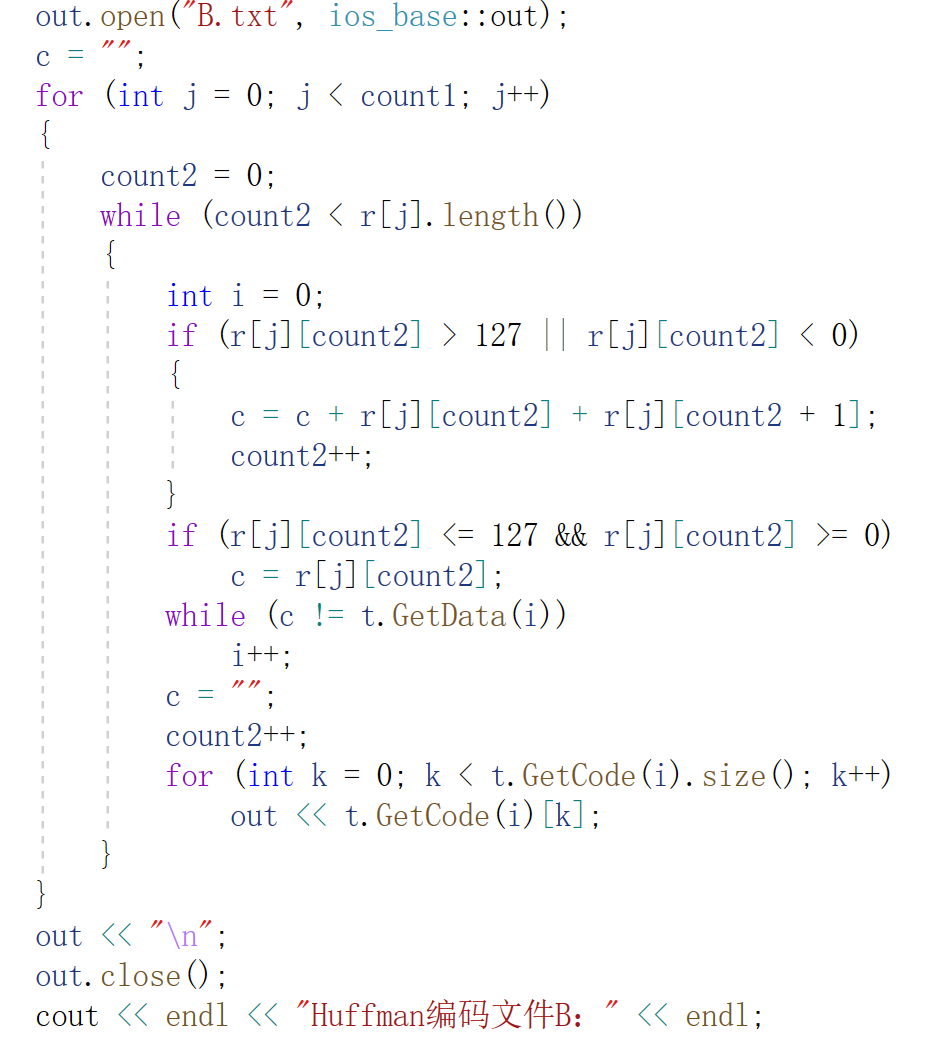


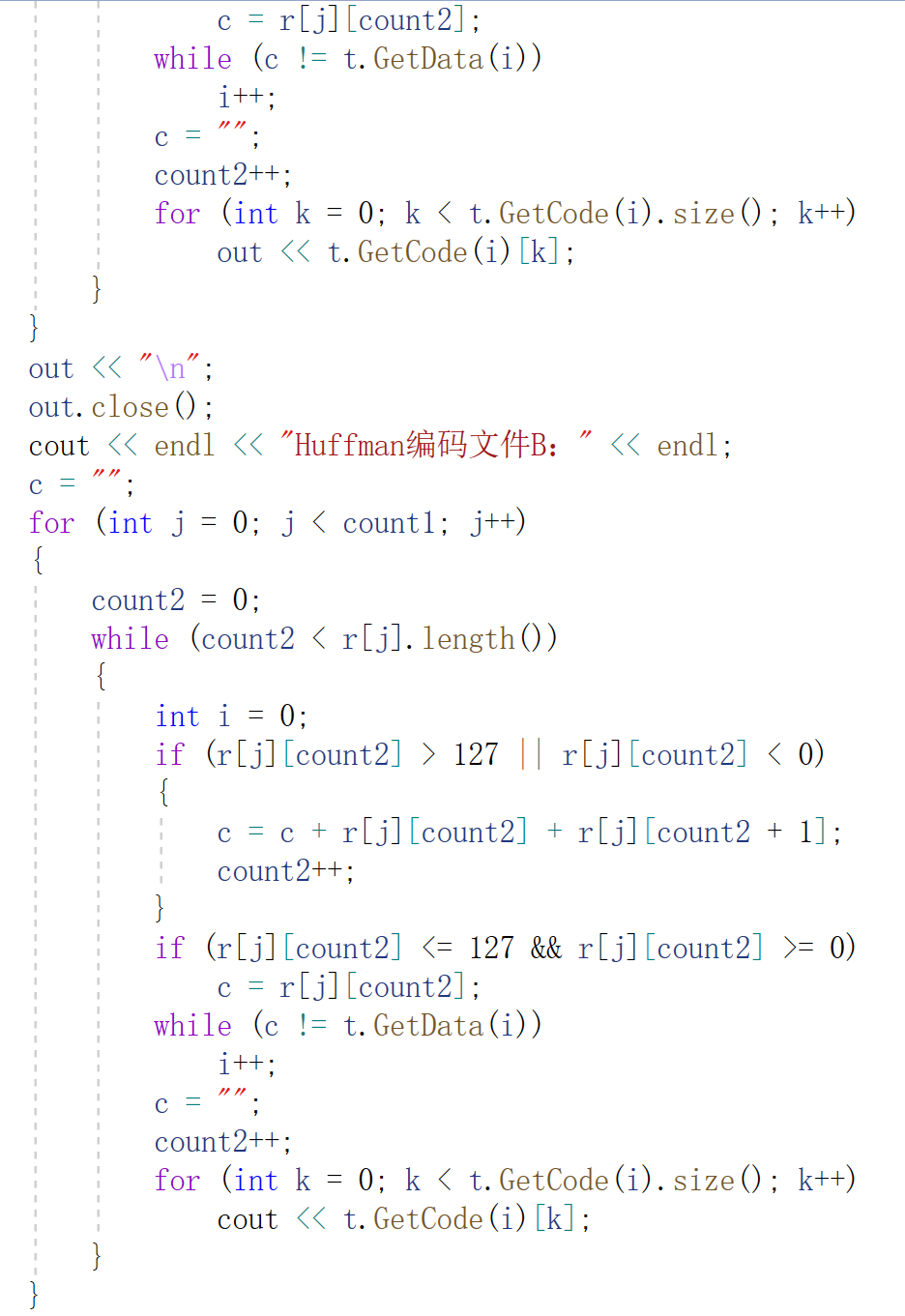


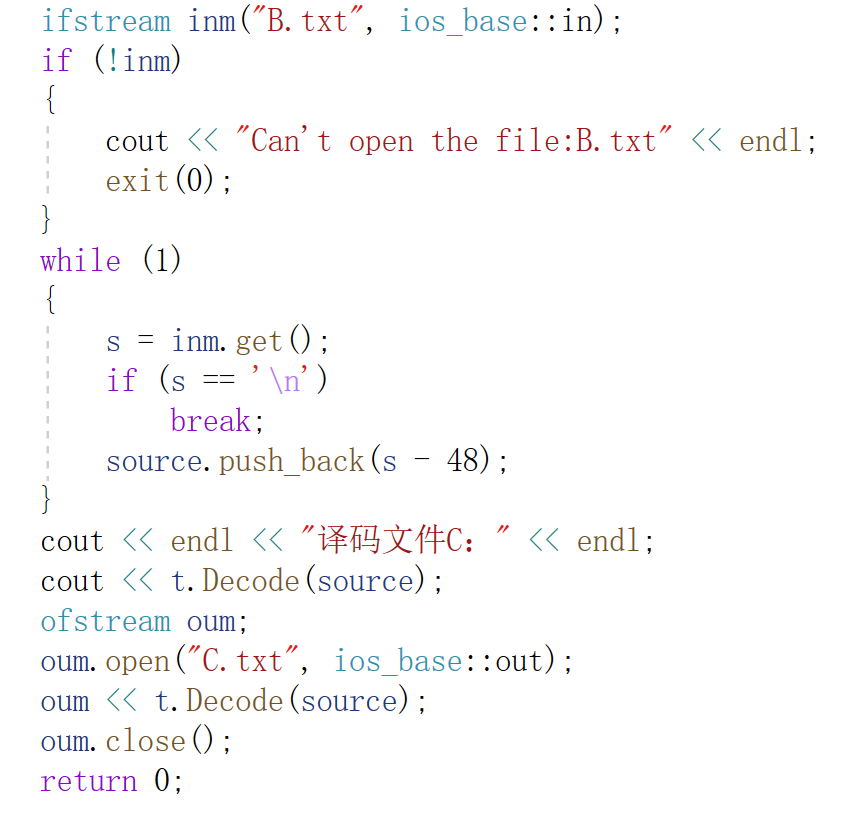












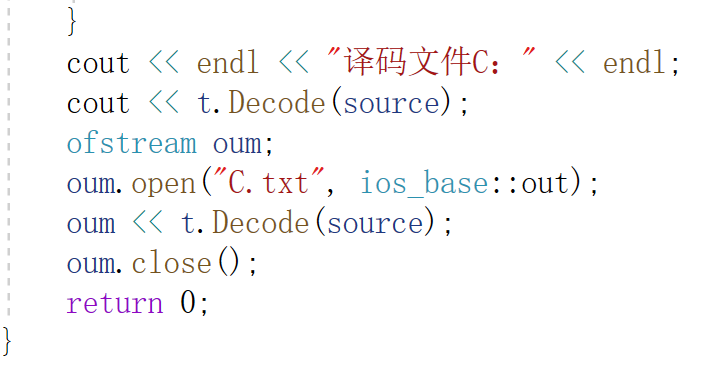
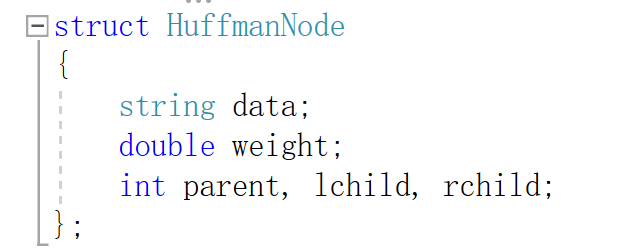


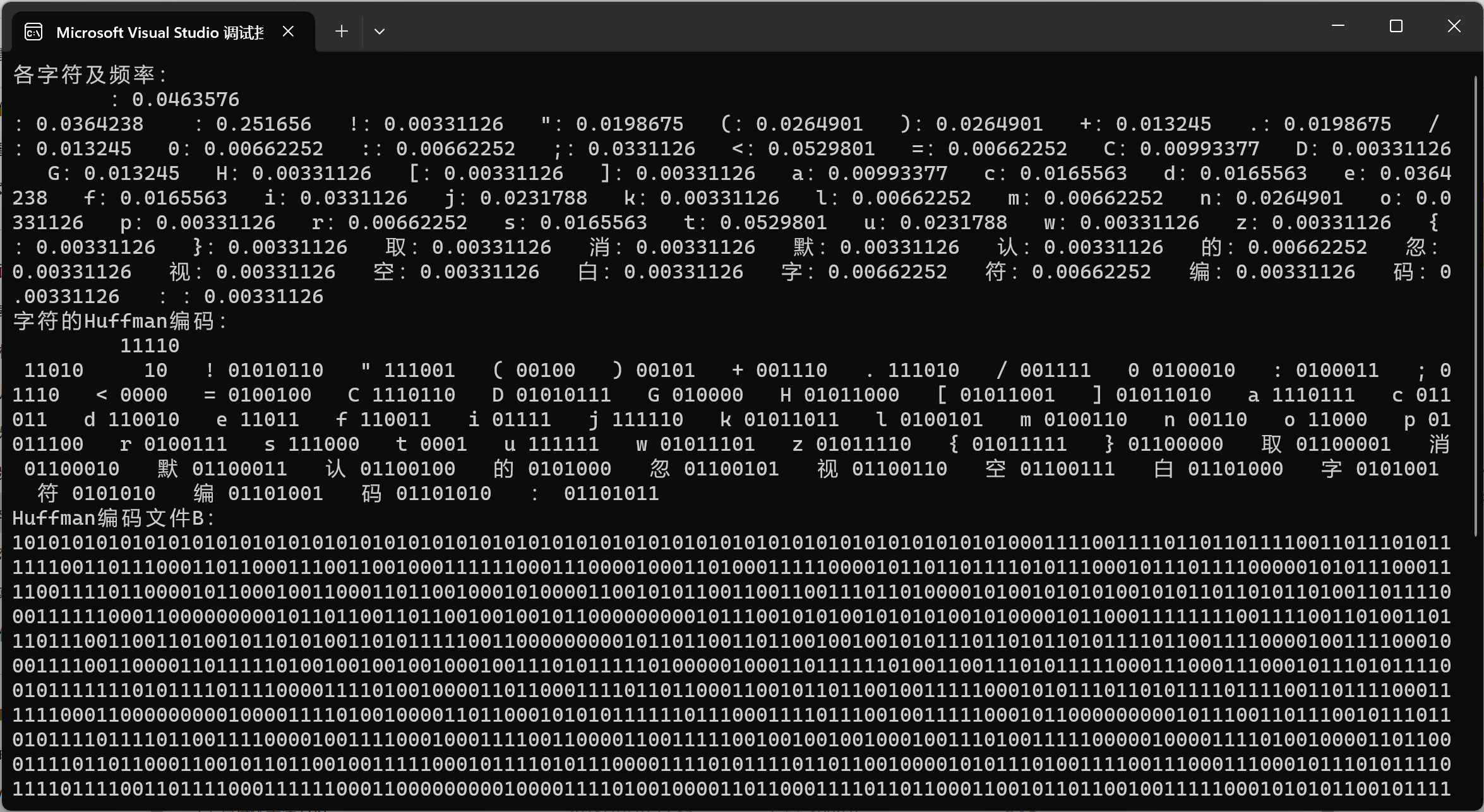
表3.2 HuffmanNode结构体

|  |  |
| --- | --- |
| 成员 | 作用 |
| string data; | 记录字符数据 |
| double weight; | 记录字符频率 |
| int parent, lchild, rchild; | 记录结点的父亲结点和孩子结点 |

相关源代码



（四）程序测试案例





三、收获与体会

数据结构课程设计是一次对我们C++语言学习的综合检验，它要求我们将所学的C++语言代码进行综合运用，设计出一个简单的系统，这不仅是对我们基础知识的检验，还考验我们的动手动脑能力。程序的基本功能较为齐全，但还是不够完美，有很多可以优化的地方。

回顾这次课程设计，使我感触颇多。很多的知识只是停留在基本的应用阶段，当需要结合使用时，往往出现无从入手的情况。在数据结构课程设计的过程中，让我深深体会到了这点。一些细节的地方没有看清楚，或者思考欠妥，就无法让系统运行，而当一系列调试后，程序能够运行。

总体来说，这次课设对我自己是一次很好的锻炼时机，不仅把以前学到的知识温习了一遍，还学习到了很多新知识，而且还提高了敲代码的能力。对设计中遇到的问题，通过仔细思考和查阅资料，使得问题得以解决，对自己以后独立思考解决问题的能力也有很大帮助。

当然，这一切体悟都离不开老师的悉心栽培。在此表达不尽的感激之情，并将牢记教诲，勤勉奋进。

四、参考文献

[1]-1073741571（0xc0000cfd）代码块c出错++[EB]

<https://www.5axxw.com/questions/content/dkwb77>

[2]警告C6262[EB].2023

<https://learn.microsoft.com/zh-cn/cpp/code-quality/c6262?view=msvc-170&f1url=%3FappId%3DDev16IDEF1%26l%3DZH-CN%26k%3Dk(C6262)%26rd%3Dtrue>

[3]随风而去飘飘飘. exited with code -1073741571[EB].2011

<https://blog.csdn.net/vblittleboy/article/details/6613815>

[4] looox007.面向对象程序设计——指针[EB].2022

<https://blog.csdn.net/Liyolo007/article/details/124651023>

[5]小宋想站起来.C++读取文件输入流读入空格换行[EB].2019

<https://blog.csdn.net/songziyuan_/article/details/97949927>

[6]软件技术爱好者.C++读取文本文件中的汉字乱码情况原因及解决[EB].2023

<https://blog.csdn.net/cnds123/article/details/131062643>

[7]独上高楼望尽天涯路. C++读入文件中的中文字符[EB].2020

<https://blog.csdn.net/qq_45045793/article/details/109632833>

[8]Cherries Man.字符编码详解及利用C++ STL string遍历中文字符串[EB].2023

<https://blog.csdn.net/myf_666/article/details/128604637>