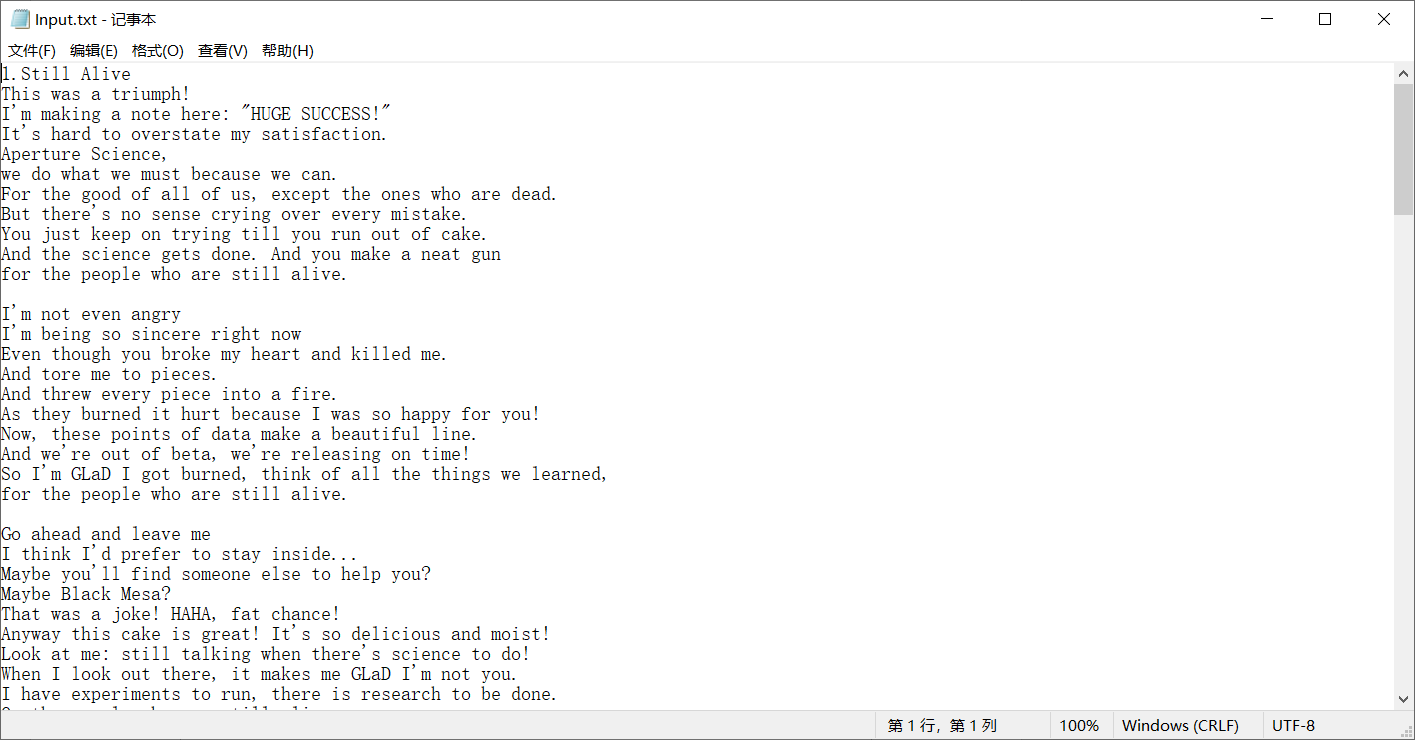
实验测试

注：测试所使用的输入文本文件请查看“程序源代码及部分文件”目录下实验主体、思考部分中Input.txt，程序输出的文件也包含在此目录下。

**Ⅰ.实验主体部分程序测试过程及结果展示**

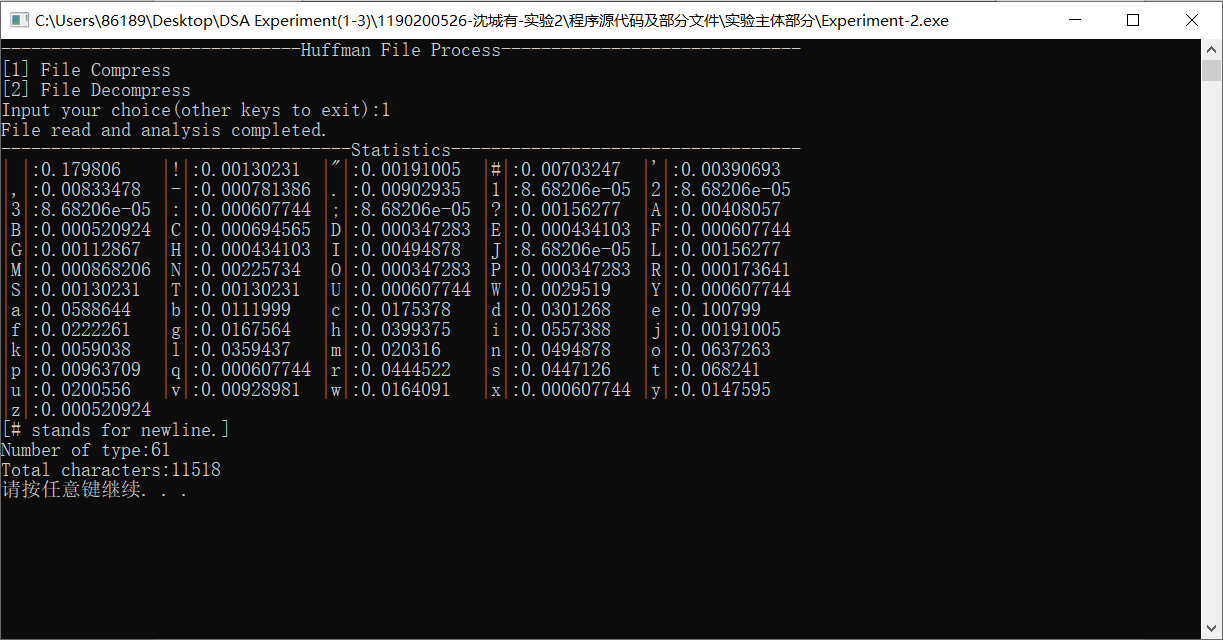
注：此部分主要是要求1-5的实现，并附加实现了01字符串与二进制01间转换以达到真正压缩文件的目的，增强了实际应用性。

1、读入文件并统计字符及其使用频率信息

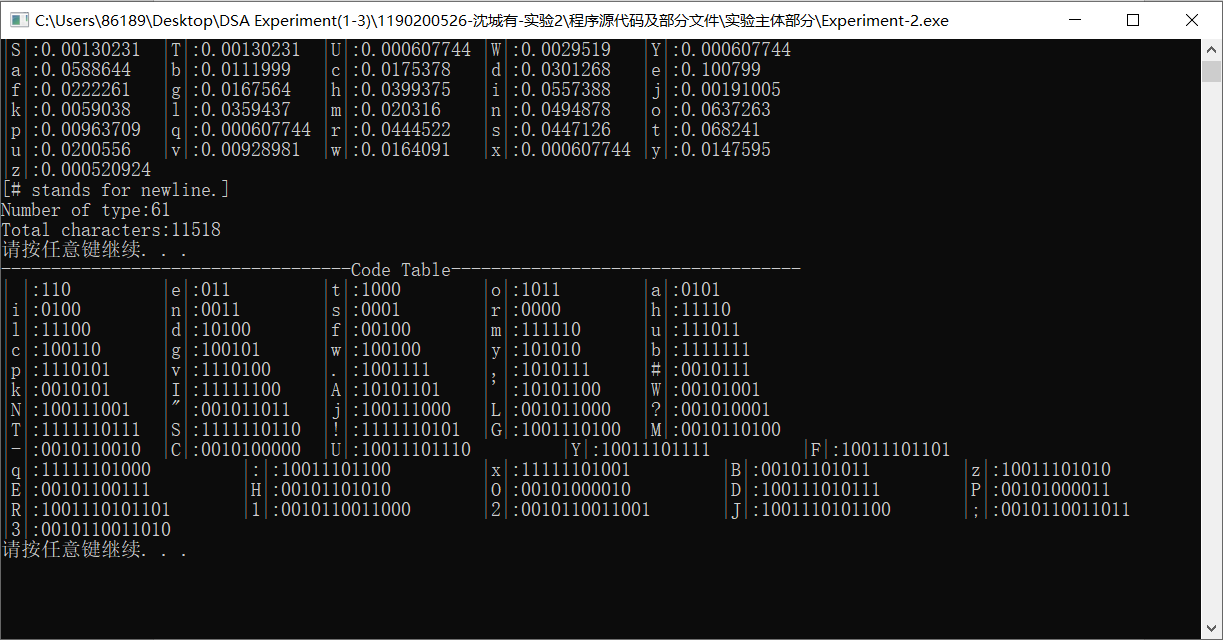
下图为读入文件内容展示。

下图为统计的字符及其频率信息。

其中“#”代表换行符，信息使用制表符尽量对齐。

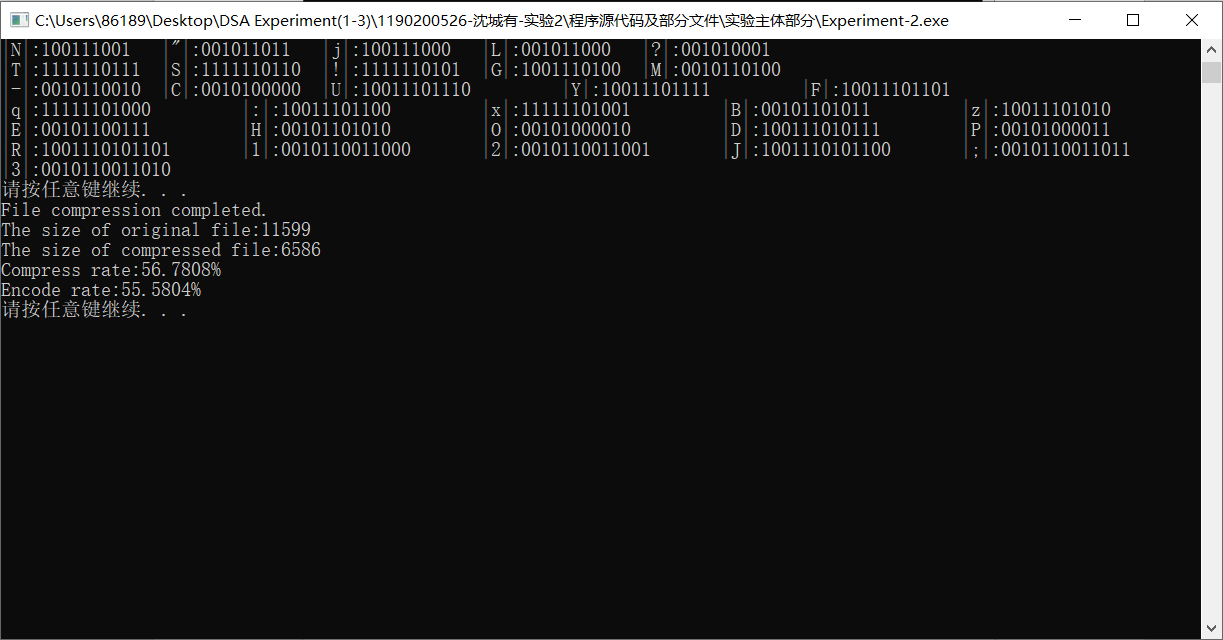


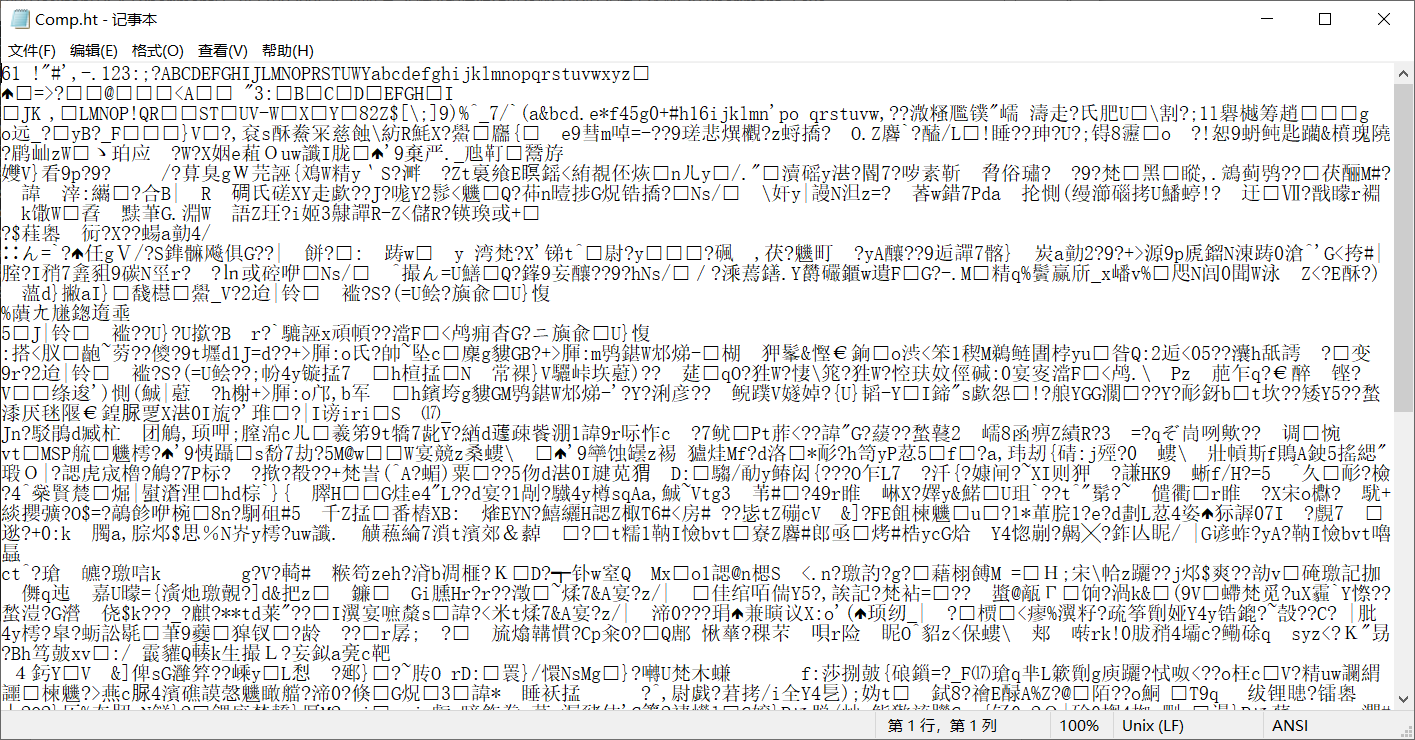
2、构造哈夫曼树并显示编码表



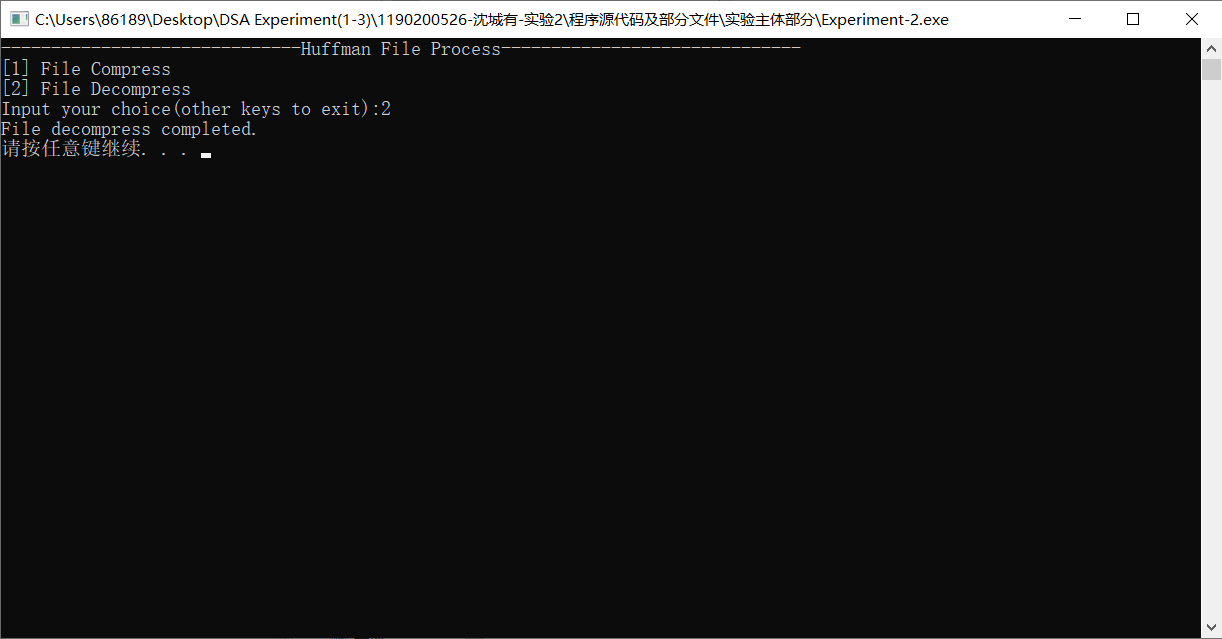
3、文件压缩并计算压缩率

下图为程序显示，数据中第一个为原文件大小（字节），第二个为压缩文件大小（字节），第三个、第四个分别为文件压缩率及编码率。

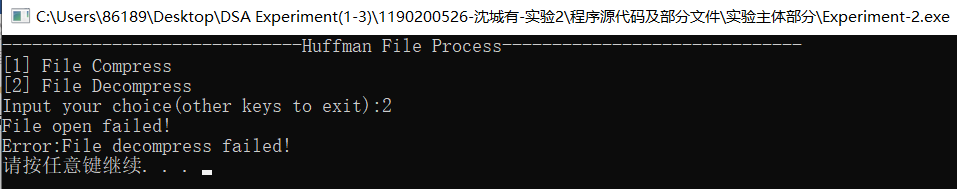


下图为编码后的文件，为真正做到压缩文件并脱离原文件及程序中暂存哈夫曼树进行解码，在文件开头写入了哈夫曼树并采用01串转二进制实现减少空间占用，故此处展示为乱码（实验思考部分展示了正常写入的编码文件）。

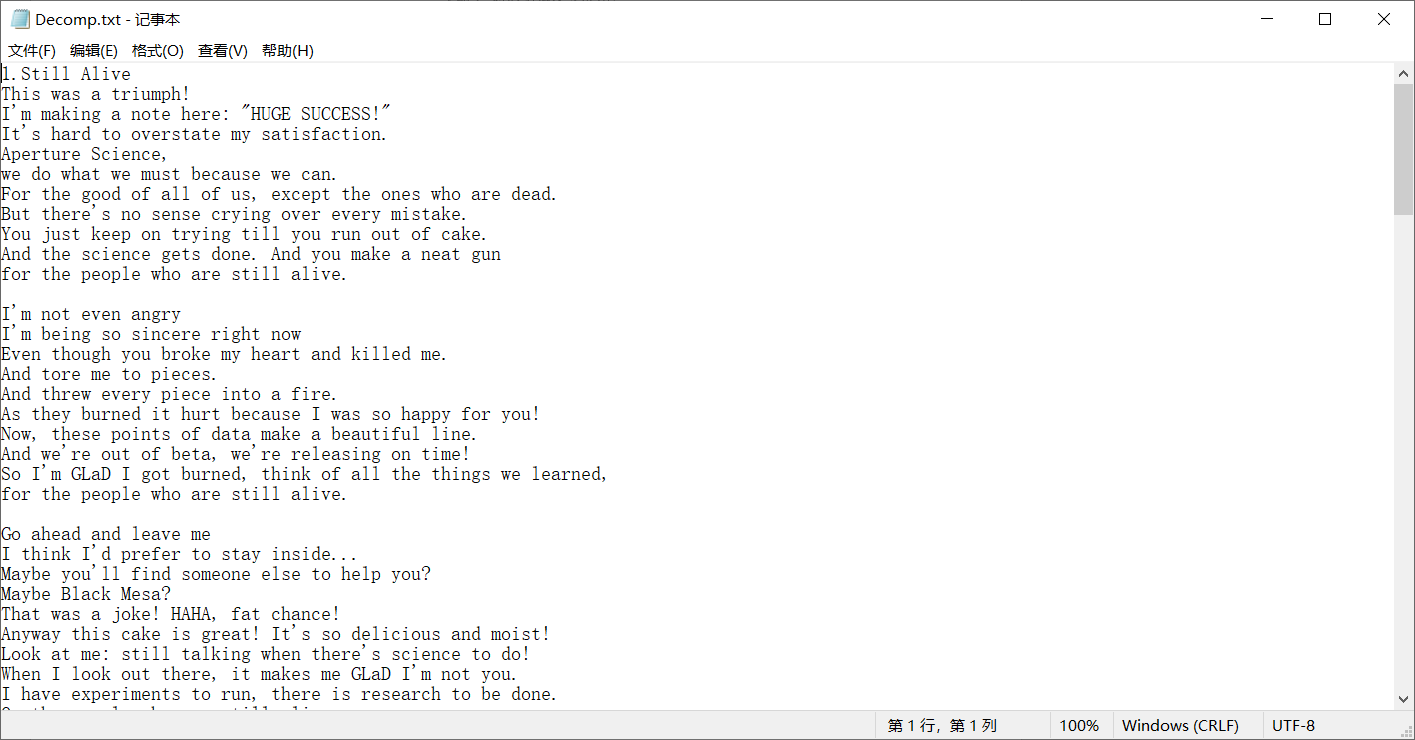
4、文件解码

图为文件解码成功时程序显示。

图为未找到或无法打开编码文件时程序显示。



5、解码后文件展示

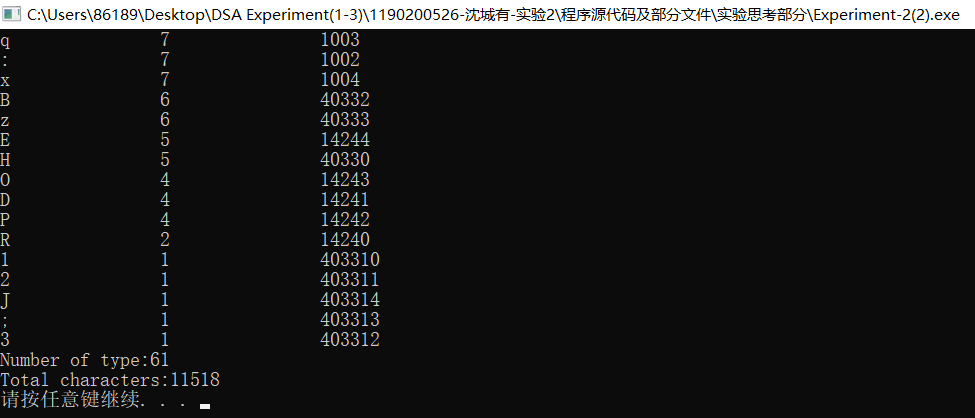
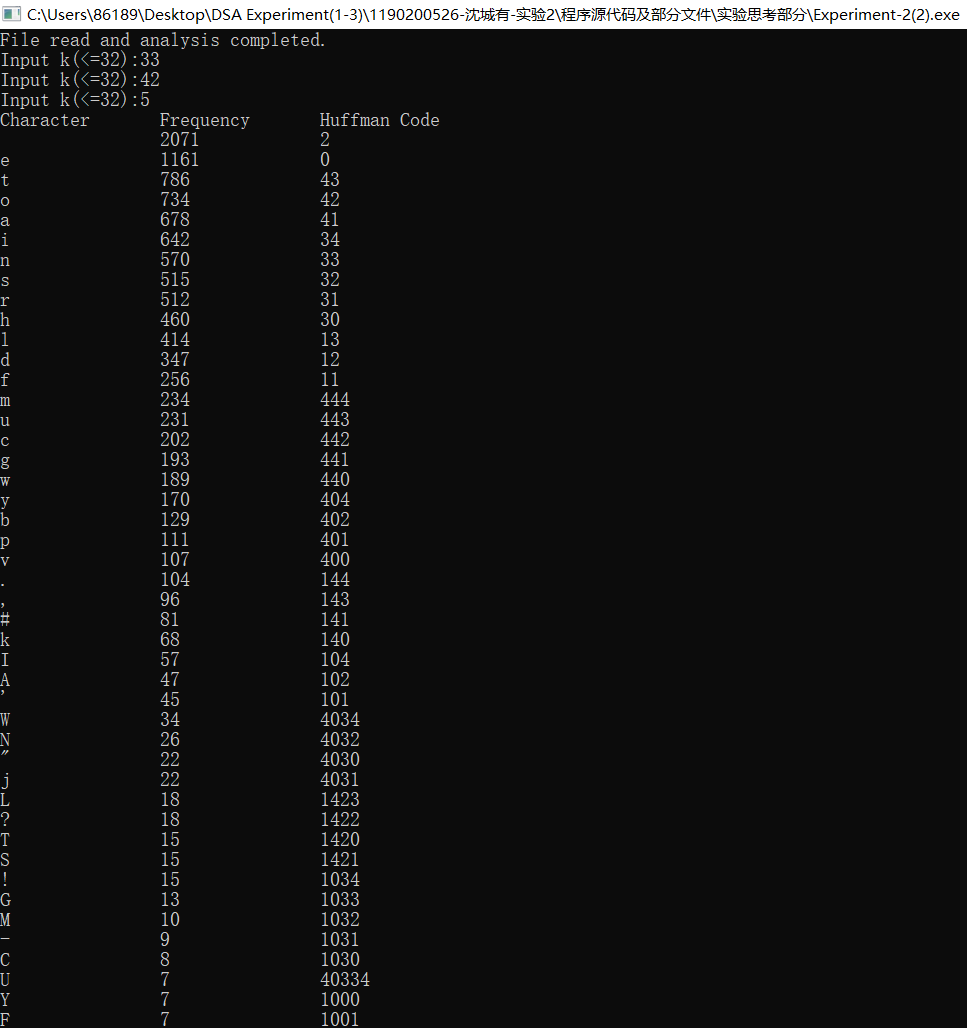


**Ⅱ.实验思考部分程序测试过程及结果展示**

注：此部分主要是K叉哈夫曼树的实现及利用最小堆优化的最小权值选取。

1、用户输入K值、文件读入及统计信息和编码表程序显示

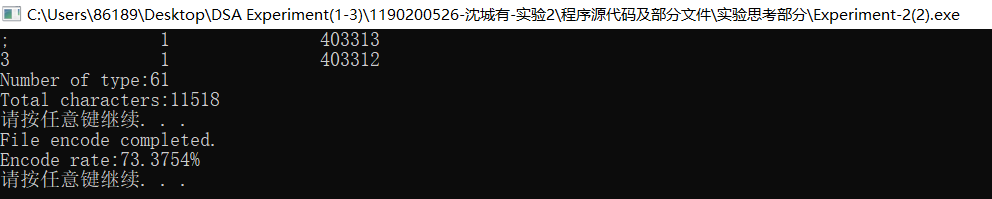
读入文件内容与主体部分一致，此处不再展示，仅展示程序部分（k=5）。



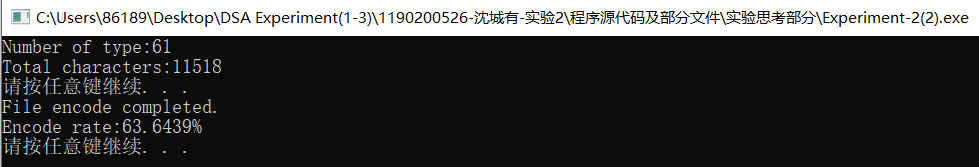
2、文件编码及计算压缩率

这里测试了三组K值，并分别计算了压缩率。

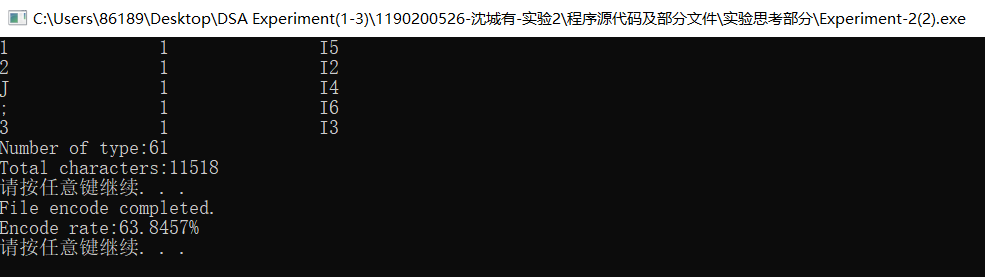
(1)K = 5:



(2)K = 13:



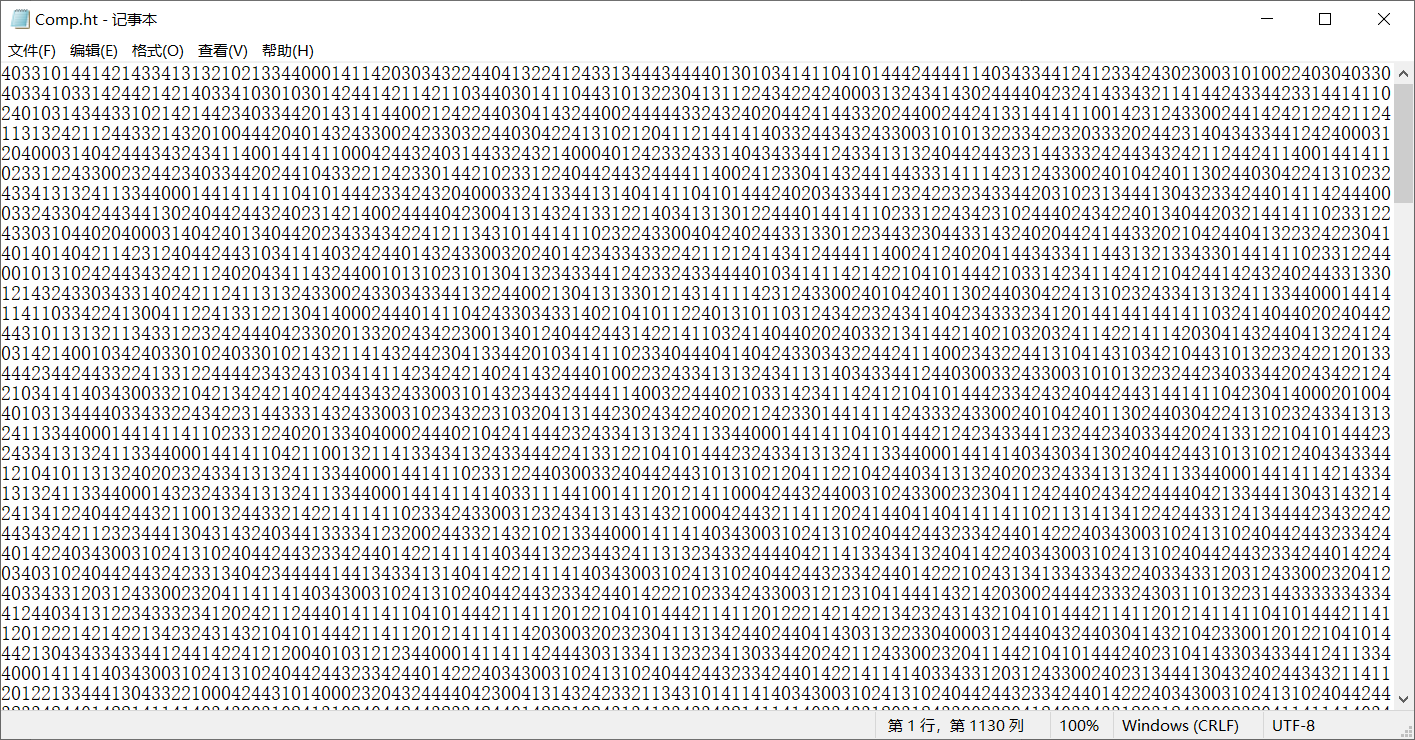
(3)K = 32:



3、编码文件展示

此处展示2中三种K值对应编码文件

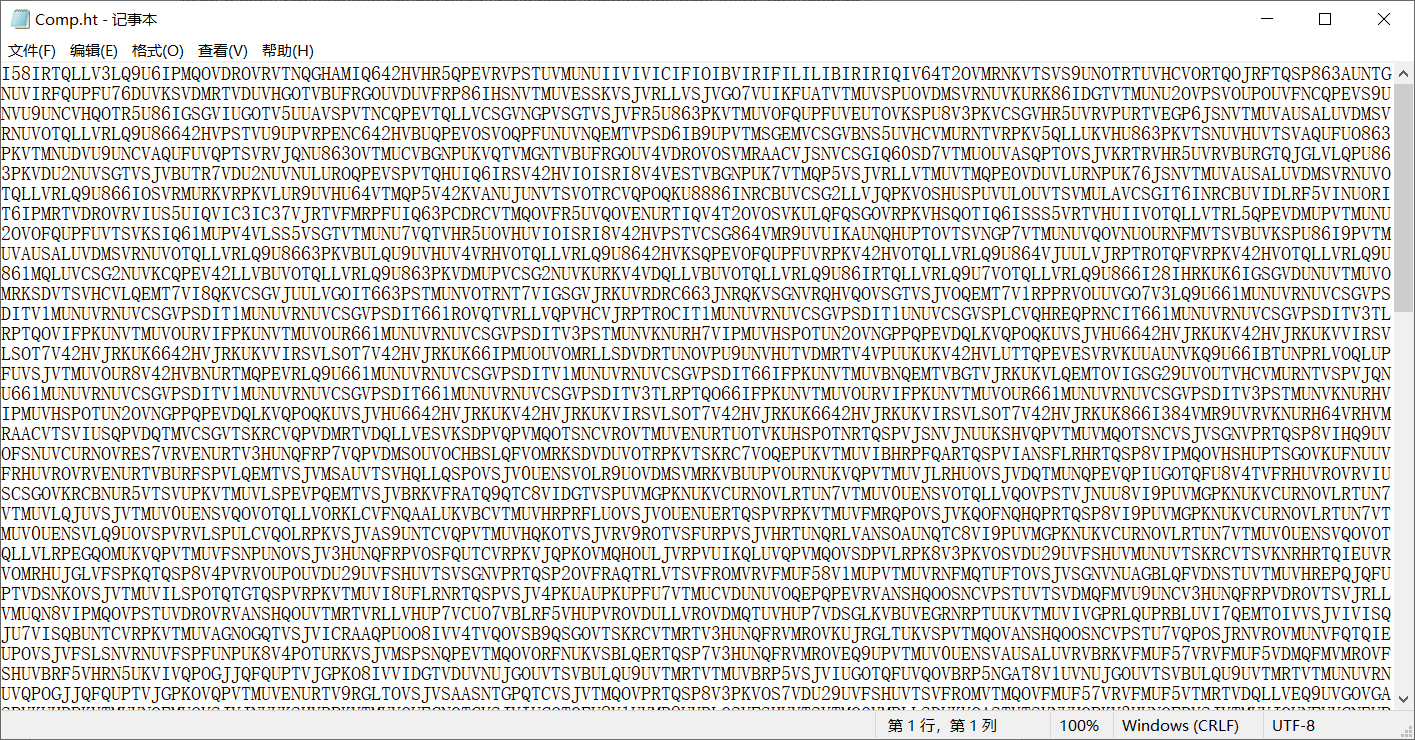
(1)K = 5:



(2)K = 13:



(3)K = 32:



此后译码与实验主体部分效果一致，不再展示。