思路

总体设计思路：

采用自动机的思想，在分析每个单词时根据其读入的首字母判断该单词可能的类型并设置当前的分析状态，包括开始分析状态、分析是标识符还是关键字状态、分析常数状态、分析注释状态、分析引号间的字符串状态、分析完成状态等。根据不同的状态进入不同的分析程序中，在不同的分析程序中根据输入继续分析，如果在可能的状态中输入合法则保留当前状态继续读入单词的字符，如果不合法就对error进行处理。当单词所含的所有字符均被读入并分析后根据分析结果可得到该单词的最终类型，重复此过程直到要分析的程序文件全部读入。单词类型包括关键字、标识符、数字、引号间的字符串、分隔符、运算符等。对于识别出的错误和注释，在另外的文件中单独展示。

设计思路分解：

常数：对于每个单词，如果第一个读入的字符为数字则进入判断该单词是否为常数的程序。常数中分了三种情况，可以处理整数、小数和指数。在读入字符的过程中如果读入小数点则记录当前读入小数点个数为1，超前读入下一个字符，如果不为数字则为error；如果读入的为E，则超前读入，如果E之后的为数字为正常情况，如果E之后读入的是+或者-，则再次超前读入，如果为数字，则为常数的正常情况，如果不是数字则为error，其余情况均为error

标识符和关键字：对于每个单词，如果第一个读入的字符为字母或者下划线则进入判断该单词是否为标识符或者关键字的程序，对剩余字符进行读入，如果均为字母数字或者下划线则为标识符或者关键字，将读入的单词与预设的关键字表进行匹配，如果能够与其中一项匹配则为关键字，否则为标识符类型。并将其加入标识符表并记录其在表中的位置。

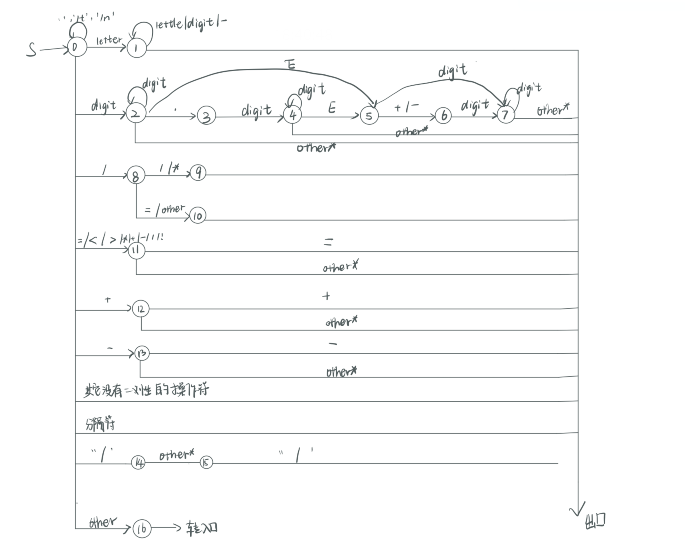
操作符：将读入符号与预设的操作符表中的项进行匹配，在识别有两个字符的操作符时需要超前读入

分隔符：将读入符号与预设的分隔符表中的项进行匹配，在识别有两个字符的分隔符时需要超前读入

引号间的字符串：对于每个单词，如果第一个读入的字符为”或者’则进入判断是否为引号间的字符串的程序，对文件中剩余字符进行读入，如果读入了”或者’则为正常状态，如果一直到文件中的字符全部读出，也没有再读入”或者’，则为error

注释：对于每个单词，如果读入的第一个字符为/则超前读入下一个字符，如果为\*或者/则进入判断是否为注释的程序。如果读入的是//则设置一个flag表示当前读入的可能是单行注释，在判断是否为注释的程序中读入剩余字符，如果读入\n则为正常情况，识别结束，否则为error；对于多行注释，读入剩余字符，如果读入的是\*则超前读入下一个字符，如果是/则为正常情况，识别结束，否则继续读入。如果一直读到文件的末尾eof都没有再读入\*和/，则为error

识别合法单词的状态转换图



在实现程序的过程中，如果不能按照定义的规则进行识别的单词将其类型设置为error，并将其位置信息和错误在error.txt文件中进行输出

需要注意的点

1.有些字符的判断有多种情况，需要超前读入。比如”/”，这个符号可能是//，/=，/\*或者/都有可能，这时需要根据相应的情况进行超前读入

2.对于多行注释的处理，读入/\*后如果没有读入\*/则继续一直读，如果文件读完都没有这个格式才能判断错误

3.对于常数类型的判断比较复杂，涉及到小数、指数、整数等形式的判断，中间需要一系列的超前读入。

结果展示

说明：

控制台显示程序运行结果

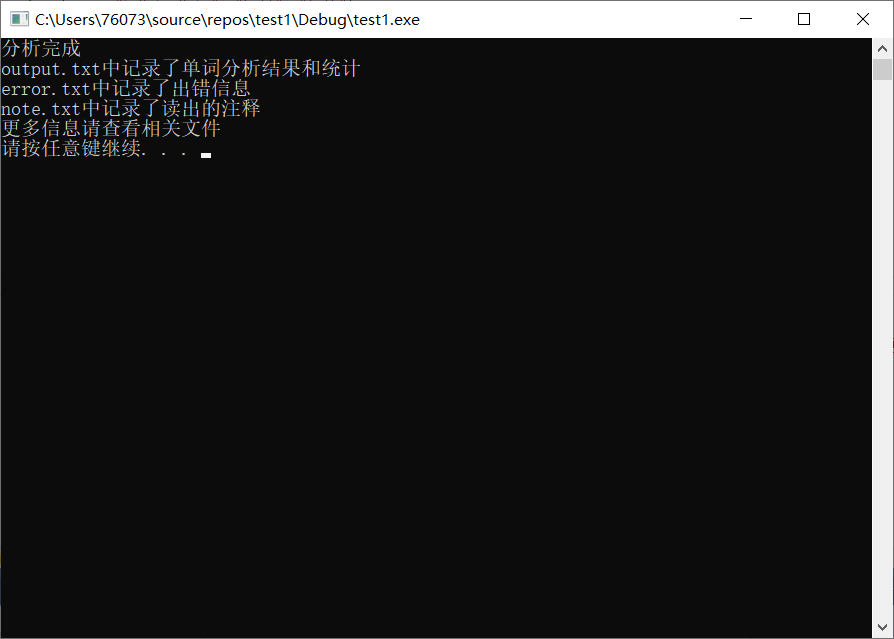
Output.txt文件输出对单词的分析结果和统计结果

Error.txt文件输出词法分析监测到的错误的行号和信息

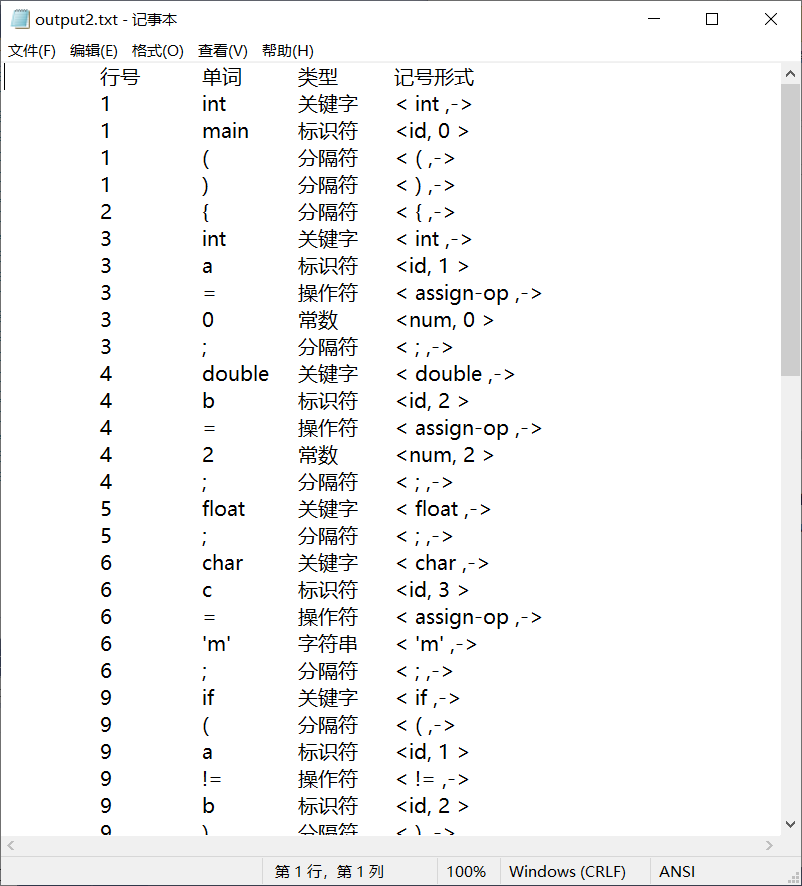
Note.txt文件输出识别出的注释，在词法分析器程序中注释被识别出来后会输出到该文件，在output文件中会被跳过

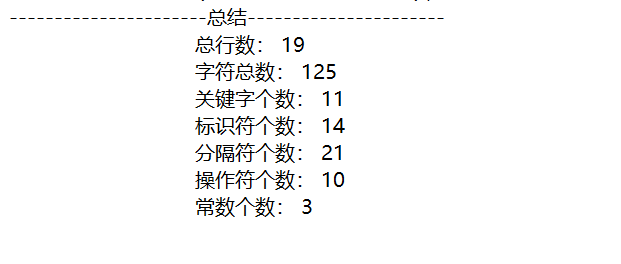
截图：

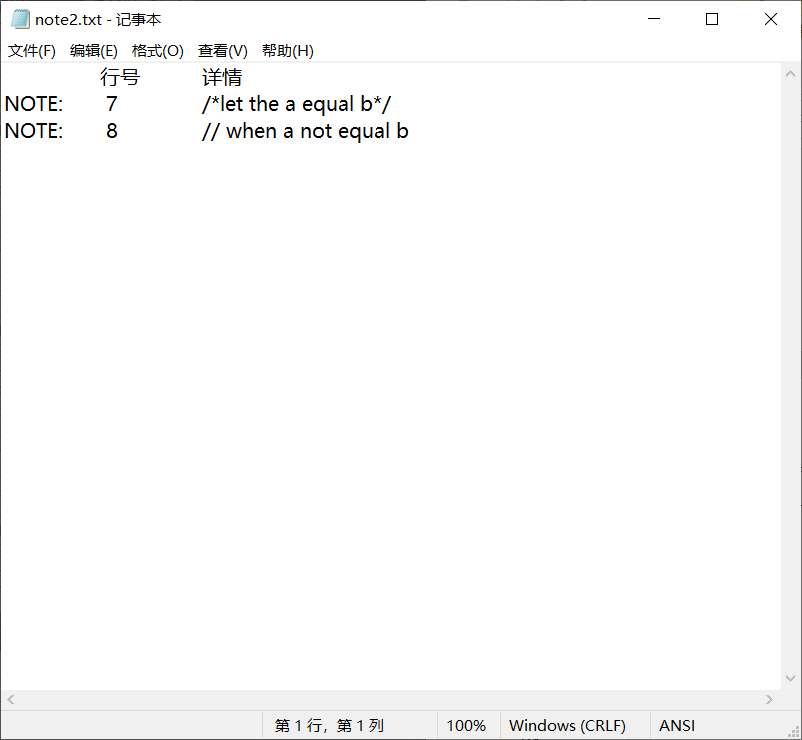
控制台：

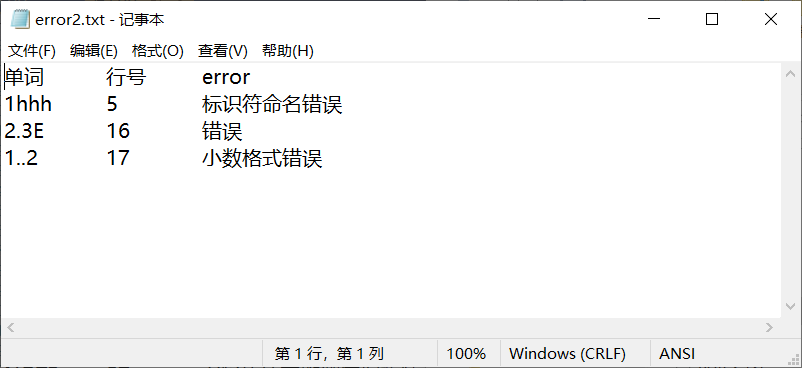


对code2进行分析的结果：









Code2代码比较少，我自己按照规则进行人工分析和统计之后结果与上述文件中输出结果一致，没有错误。

对code进行分析的结果：

