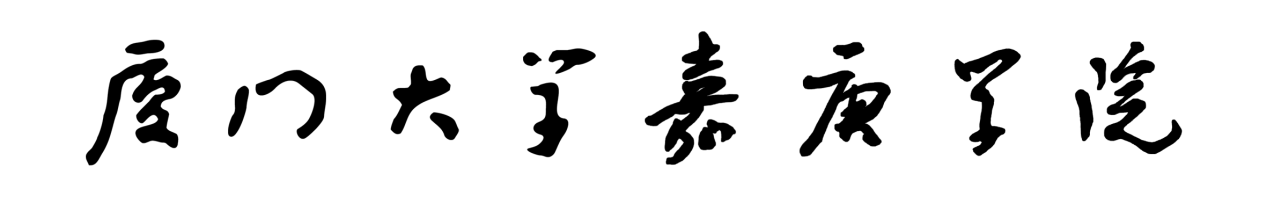
****

**(论文首页纸)**

**2018—2019 学年第 一 学期**

**课程名称： 词法分析**

**任课教师： 郭一晶**

**成员：**

**SWE16003徐麒涵**

**SWE16004黄潇**

**SWE16024连嘉华**

**SWE16035蔡志鹏**

提交日期：2018年11月6日

1. **实验目的**

通过设计编制调试一个具体的词法分析程序，加深对词法分析原理的理解。并掌握在对程序设计语言源程序进行扫描过程中将其分解为各类单词的词法分析方法。

编制一个读单词过程，从输入的源程序中，识别出各个具有独立意义的单词(token)，即基本保留字、标识符、常量、运算符、分隔符五大类，并依次输出各个单词的内部编码及单词符号自身值。（遇到错误时可显示“Error”，然后跳过错误部分继续显示）。

1. **实验分工**

徐麒涵：负责使用PHP实现

黄潇：负责使用C++实现

连嘉华：负责使用C++实现

蔡志鹏：负责使用golang实现

1. **程序说明**

程序先使用fin.open()函数读取文件test.txt中内容，在对读取的内容逐字进行匹配。程序能够识别基本保留字、标识符、常量（包括字符、字符串、实数、科学记数法、16进制数字）、运算符、分隔符。

|  |  |
| --- | --- |
| id | 内容 |
| 1 | 基本保留字 |
| 2 | 标识符 |
| 3 | 常量 |
| 4 | 运算符 |
| 5 | 分隔符 |

当识别到了以上内容时，将按照格式(id,”识别内容”)输出

函数说明：

void analysis()：显示匹配到的结果

void error()：错误处理

bool reserve()：判断token中的字符串是否在保留字表中

void getbe()：若character中的字符为空，则继续读字符，直到非空

bool letter()：判断character中的字符是否为字母

bool digit()：判断character中的字符是否为数字

void buildlist(int id)：将标识符登记到符号表中(id=2),将常数登记到常数表(id=3)

bool isEnd()：判断是否识别数字的结尾

bool isOp(char getOp)：判断是否是加减乘除符号

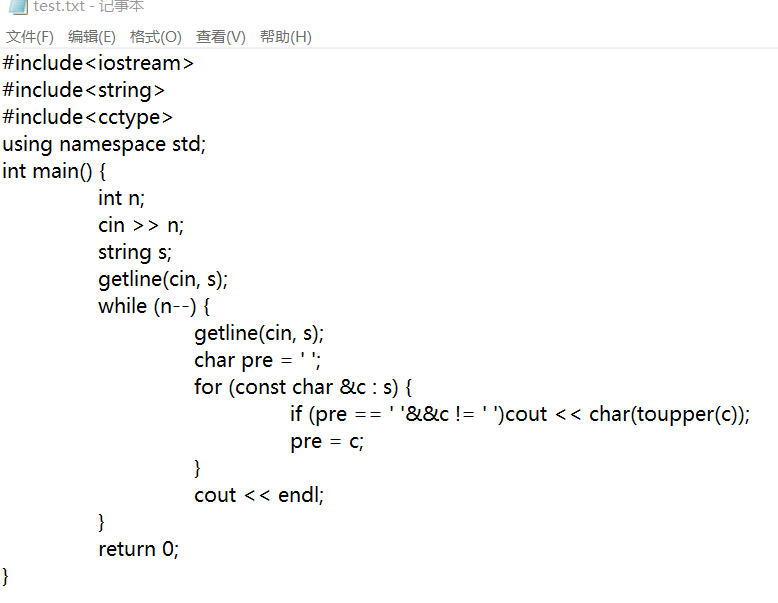
int isUnsignNum()：判断科学记数法，返回：3代表数字，4代表运算符，-1代表错误

1. **其他说明**
2. 实数的识别：若一个字符串中最多有不超过1个小数点，其余为数字，则为实数；
3. 剔除注释：
   1. /\* \*/：先判断当前字符是否为/，且下一个字符为\*，则代表注释开始，直到读取到\*/则为注释结束，如果识别不到\*/，将会**终止识别**，并输出”错误/\*\*/不匹配”
   2. //：先判断当前字符是否为/，且下一个字符为/，则代表注释开始，直到读取到换行符则为结束
4. 检查{}、()、[]的配对：当读到(、[、{的时候将其存入栈中，读到}、]、)的时候读取栈顶的括号，如果不匹配，则说明括号不匹配，程序将会输出”括号不配”并**终止识别**
5. 检查的关键字包括("try", "catch", "class", "signed", "unsigned", "char", "int", "short", "long", "bool", "size\_t","enum", "false", "true", "float", "double", "string", "typedef", "void", "struct", "define", "const", "delete", "dynamic\_cast","static", "static\_cast", "switch", "case", "break", "default", "for", "while", "do", "continue", "if", "else", "goto", "explicit","export", "extern", "new", "operator", "private", "public", "protected", "register", "return", "sizeof", "throw", "template", "this", "typeid","friend", "inline", "using", "namespace", "union", "virtual", "reinterpret\_cast", "typename", "volatile", "wchar\_t", "main")
6. 检查的运算符中，将**不识别**强制类型转化（例如(int),(double)）和条件运算符(?:)
7. 对于预处理指令（#开始的）将直接输出，并不会判断是否合法

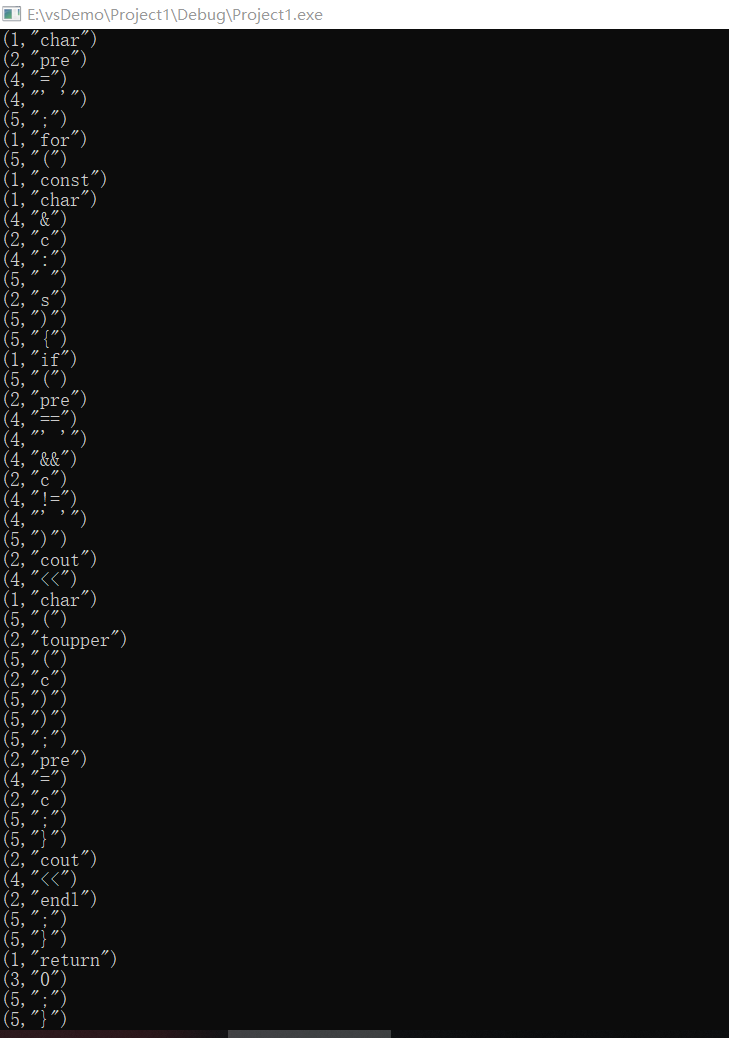
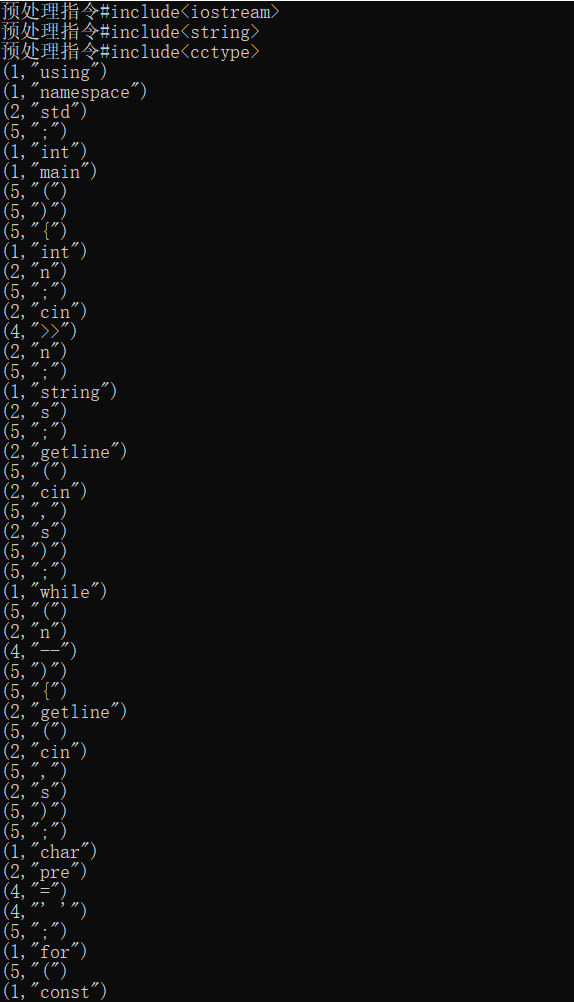
可以改进的地方:

1. 换行只针对了\r\n类的换行
2. 字符串和字符可以考虑识别合不合法
3. **运行结果截图显示**

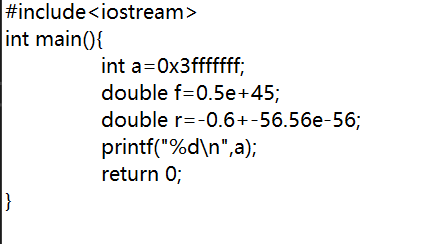
输入：



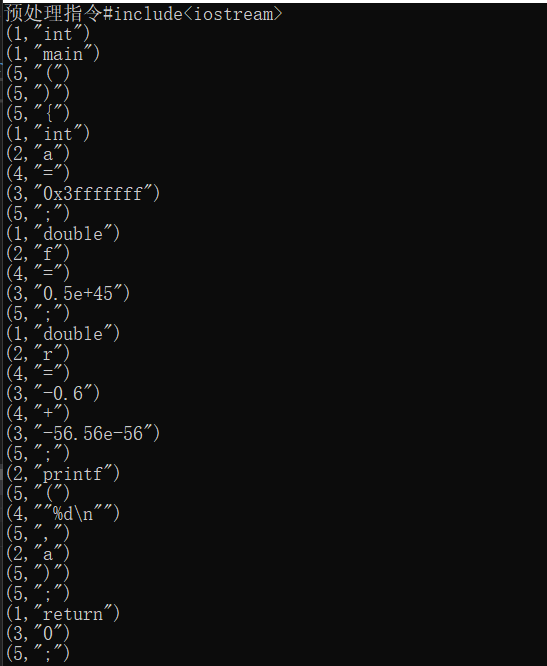
输出



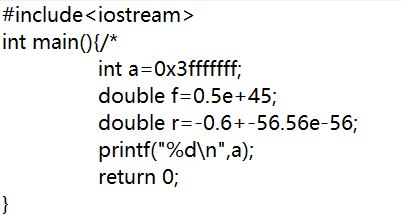
输入：



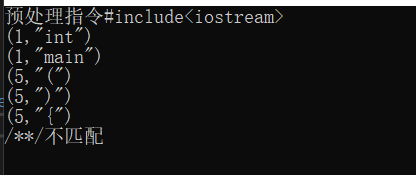
输出



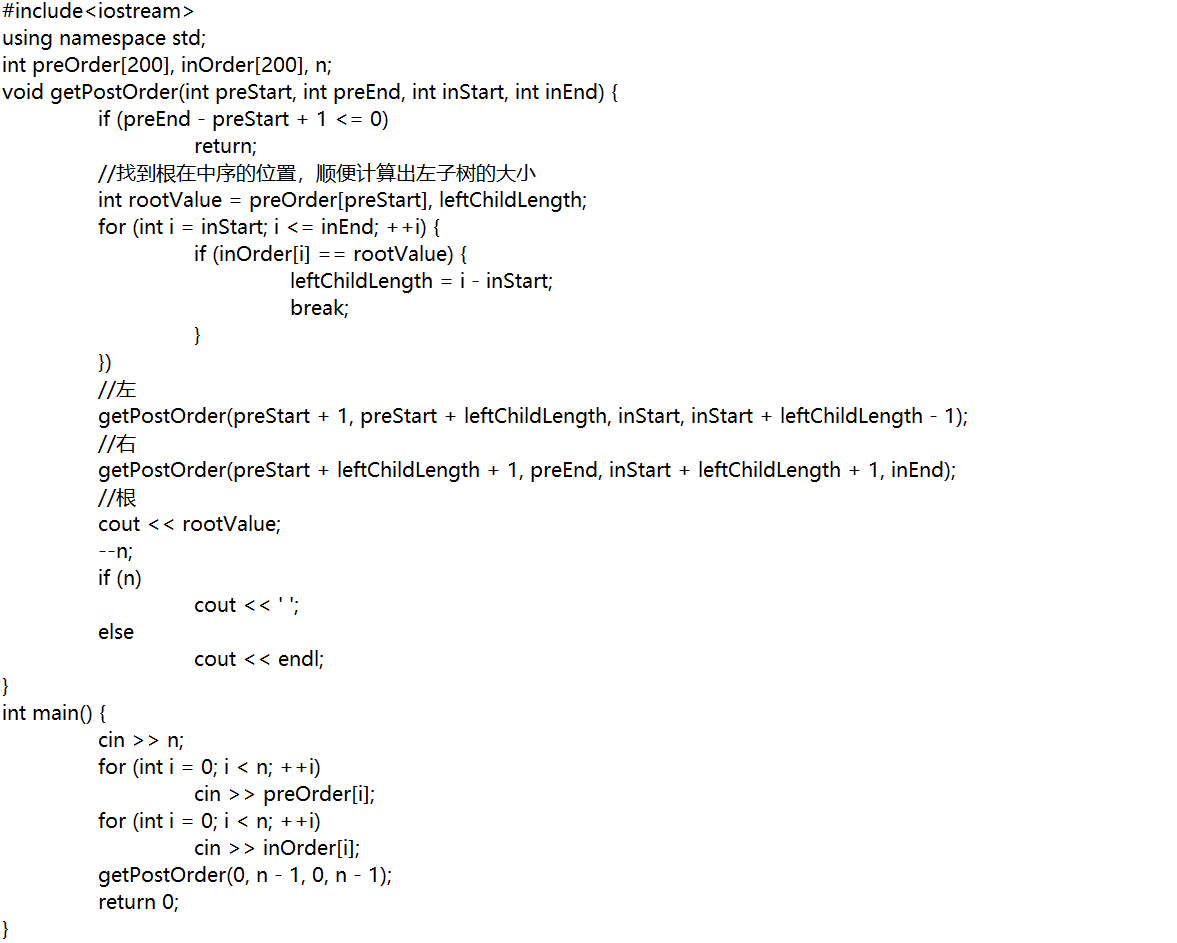
输入



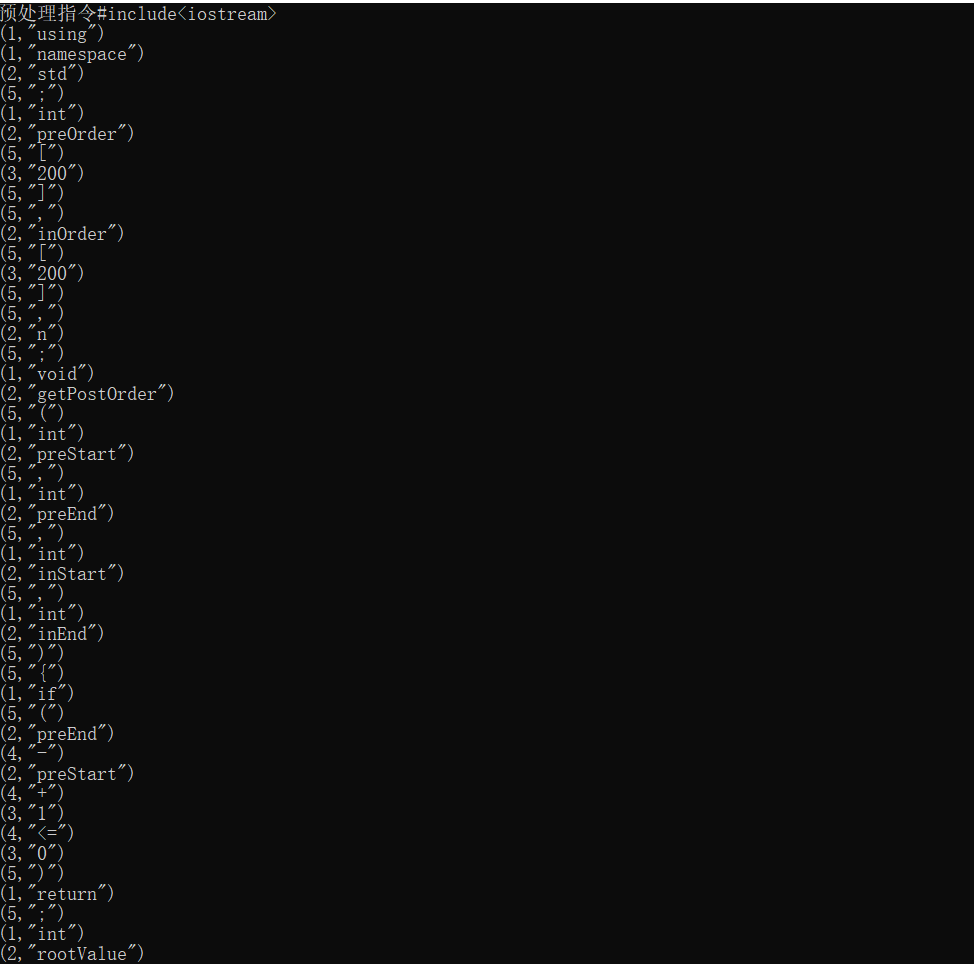
输出

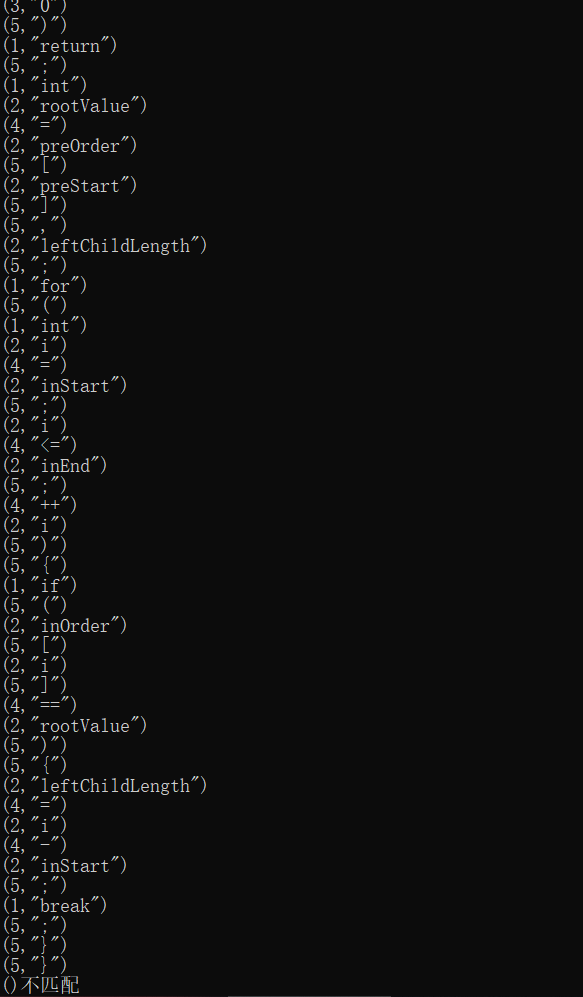


输入



输出





1. **源程序**

见analayisis.cpp

测试案例：input为前缀的文件代表输入，对应output为后缀的文件代表对应的输出