

# 词法分析器

## 一、运行和开发环境

运行环境: window 10

开发环境: QT5.14.2

## 二、能识别的单词、扩充的功能

能识别的保留字单词和运算符如下:

```
//保留字表
static char reserveWord[WORDLEN][20] = {
    "include", "using", "namespace", "std", "define",
    "main", "bool", "auto", "break", "case", "char", "const", "continue",
    "cout", "cin", "printf", "scanf", "_getch", "default", "do", "double", "else", "enum", "extern", "while",
    "float", "for", "goto", "if", "int", "long",
    "register", "return", "short", "signed", "sizeof", "static",
    "struct", "switch", "typedef", "union", "unsigned", "void",
    "volatile", "system"
};
//运算符表
static char oneOpera[ONEOPLEN] = {
    '+', '-', '*', '/', '<', '>', '=',
    '(', ')', '^', '|', '\\', '\\', '#', '&',
    '|', '%', '~', '[', ']', '{', '\\',
    '}', '.', ':', '!', '\\?'
};
static char twoOpera[TWOOPLEN][3] = {
    "<=", ">=", "==",
    "!=", "&&", "||", "<<", ">>"
};
```

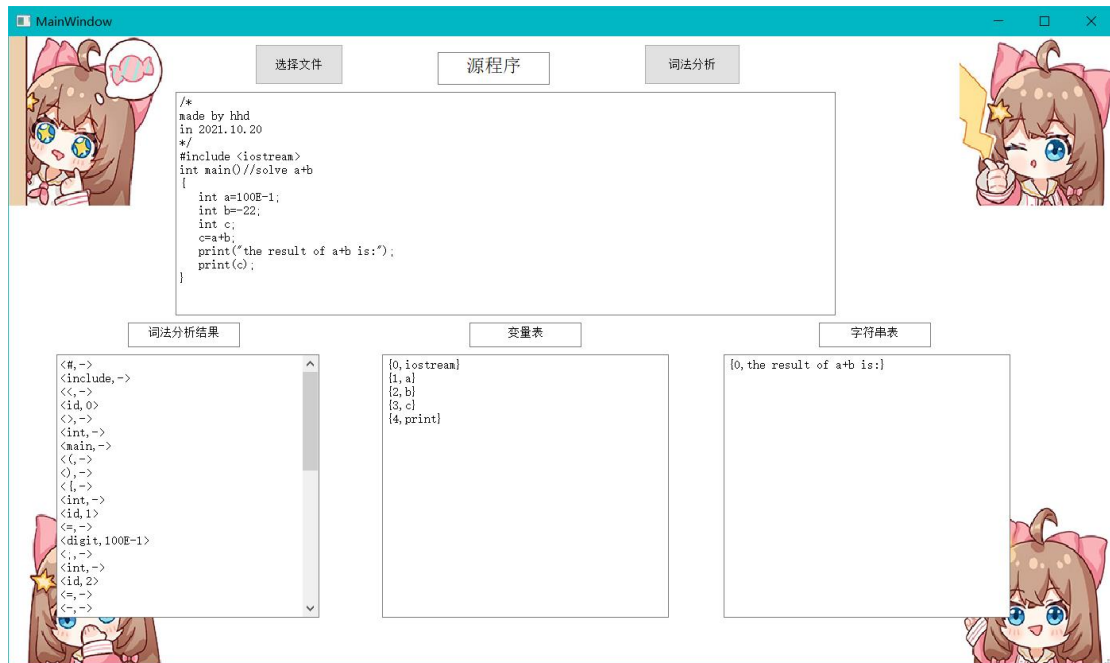
相比于实验原本要求有了一定的扩充。

以下是一些扩充功能:

**预处理程序:**

这里的进行了预处理但是没有专门打印出来。

如下的语法分析



其预处理的结果在打印台的输出如下：

```

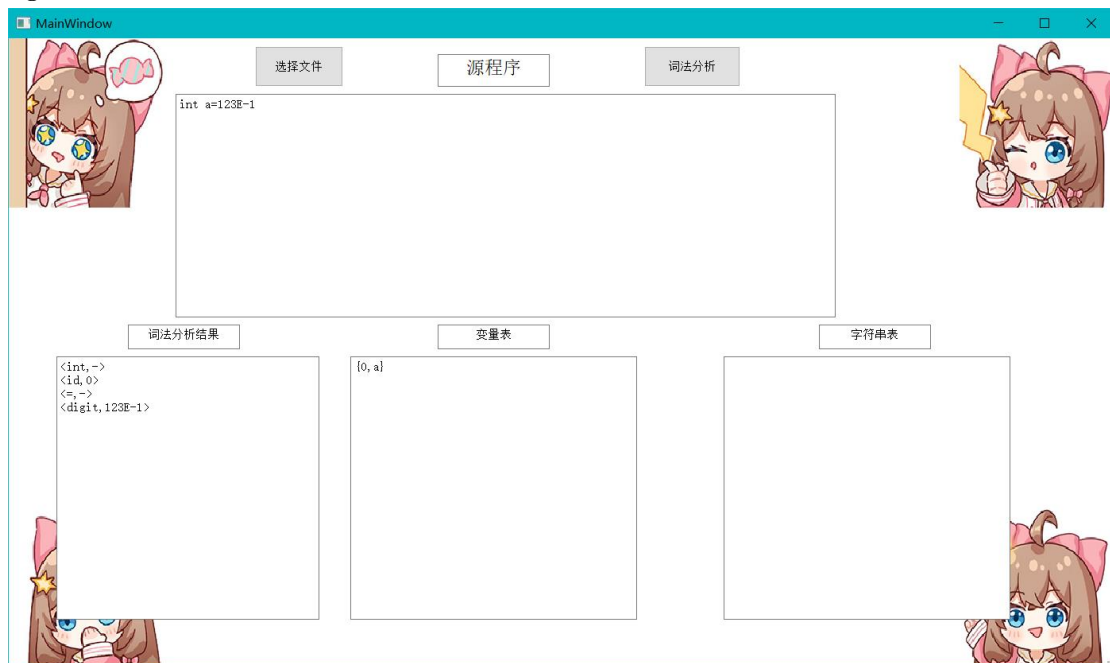
08:51:30: Starting D:\Qt\project\build-byyl-Desktop_Qt_5_14_2_MinGW_64_bit-Debug\debug\byyl.exe ...
after pretreatment:
#include <iostream>int main(){int a=100E-1;int b=-22;int c;a+b;print("the result of a+b is:");print(c);}the result of a+b is:

```

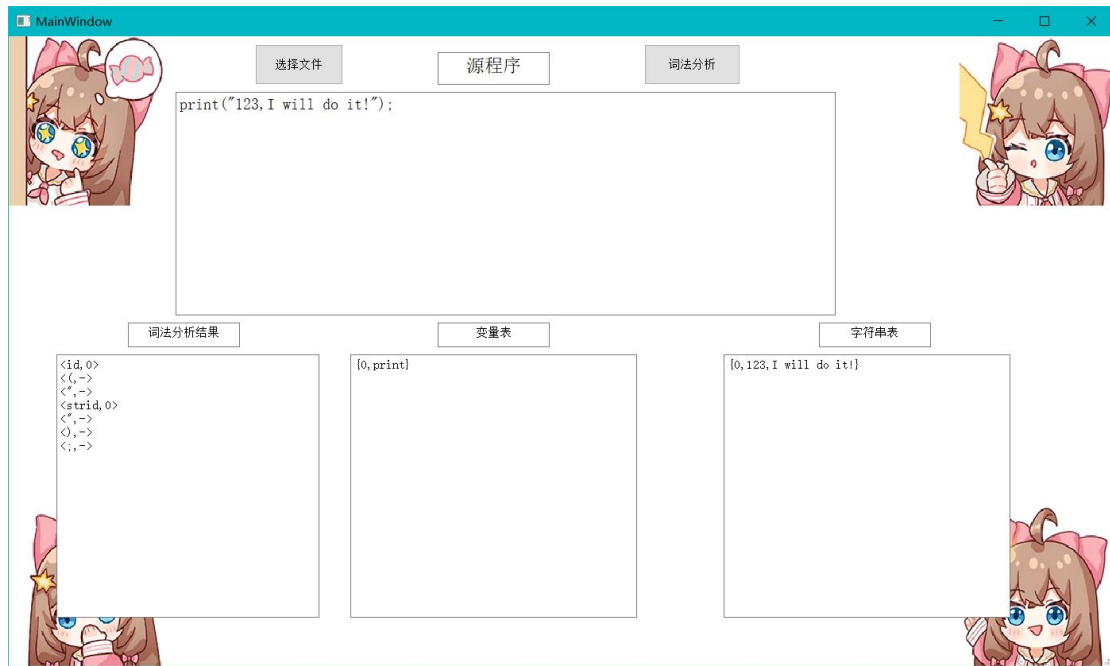
在预处理的时候消去了注释的部分，将换行删除，同时将多个空格合并为一个空格。

能识别的数字格式也进行了扩充：

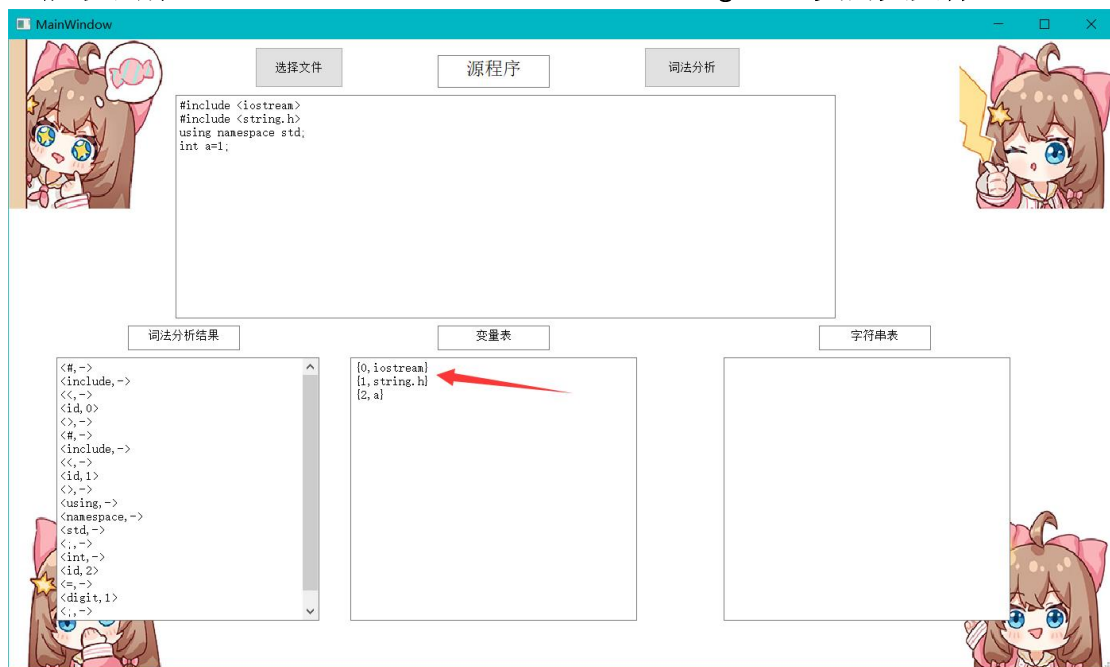
eg: 123, +123, -123, -34.56, 0.233, 1.23e3, 2.33E+4, -13.33E-4



还能够识别识别出 print(“123,I will do it!”)中“ ”内的字符串

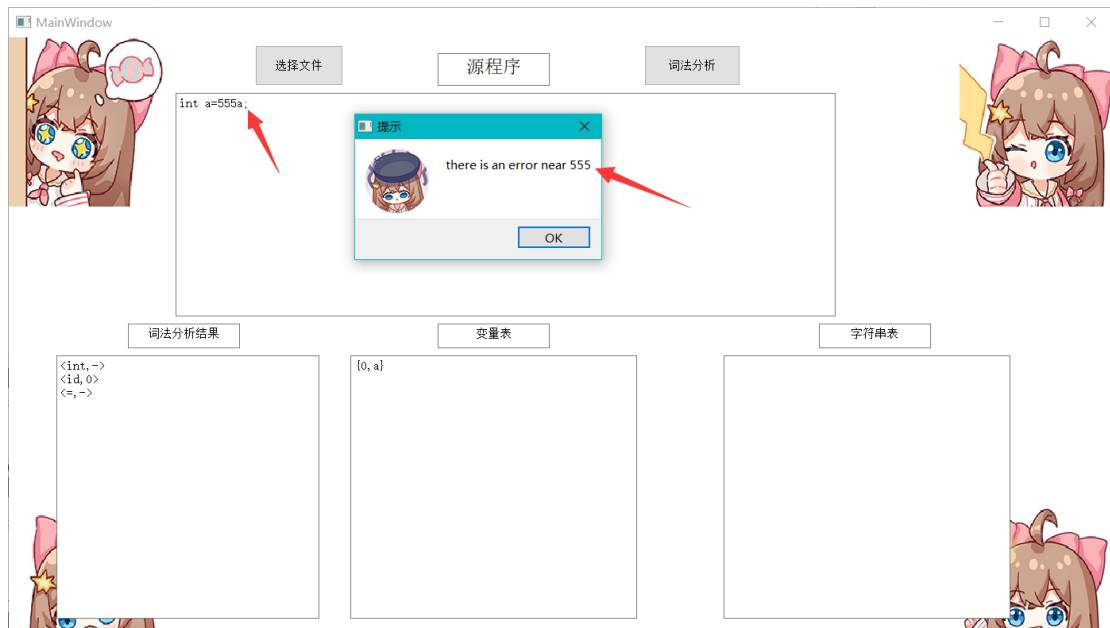


还能够识别`#include<iostream>`，`#include<string.h>`这类的头文件。



## 对于语法错误也有一定的报错能力

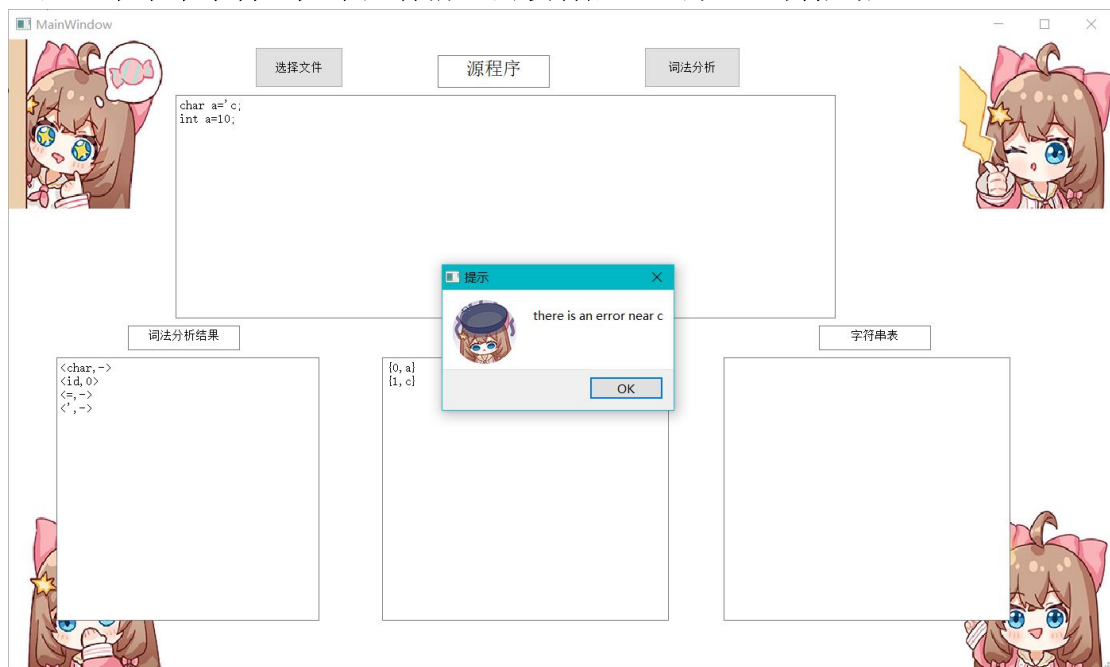
数字方面，如果是 123abc、1.23E-23cdfg 这种正规的数字后面紧邻着字母，那么就会报错，并且报错时会将近临的数字显示出来。



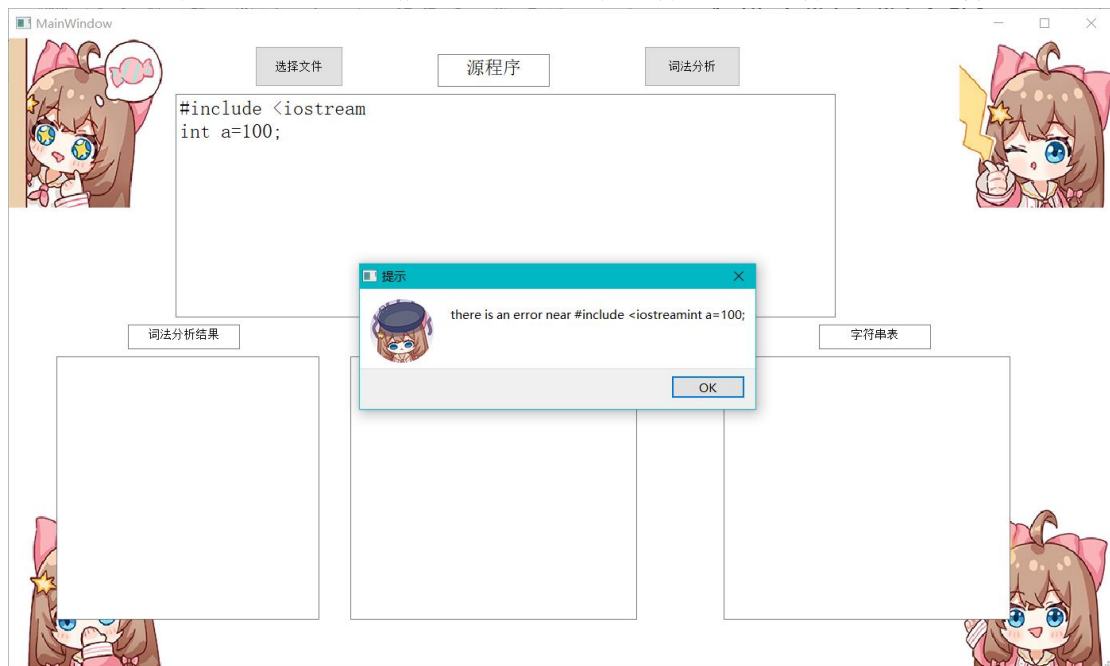
因为处理了引号内的字符串，所以也对其进行一定的错误判别，如果是只有前引号而没有后引号的话，就会报错



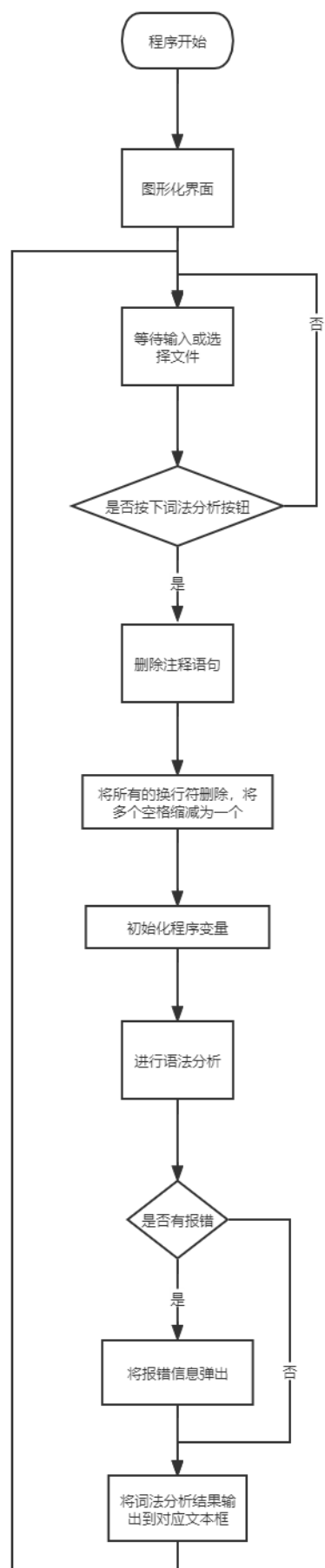
还处理了单个字符，如果只有前'而没有后'，那么也会报错。



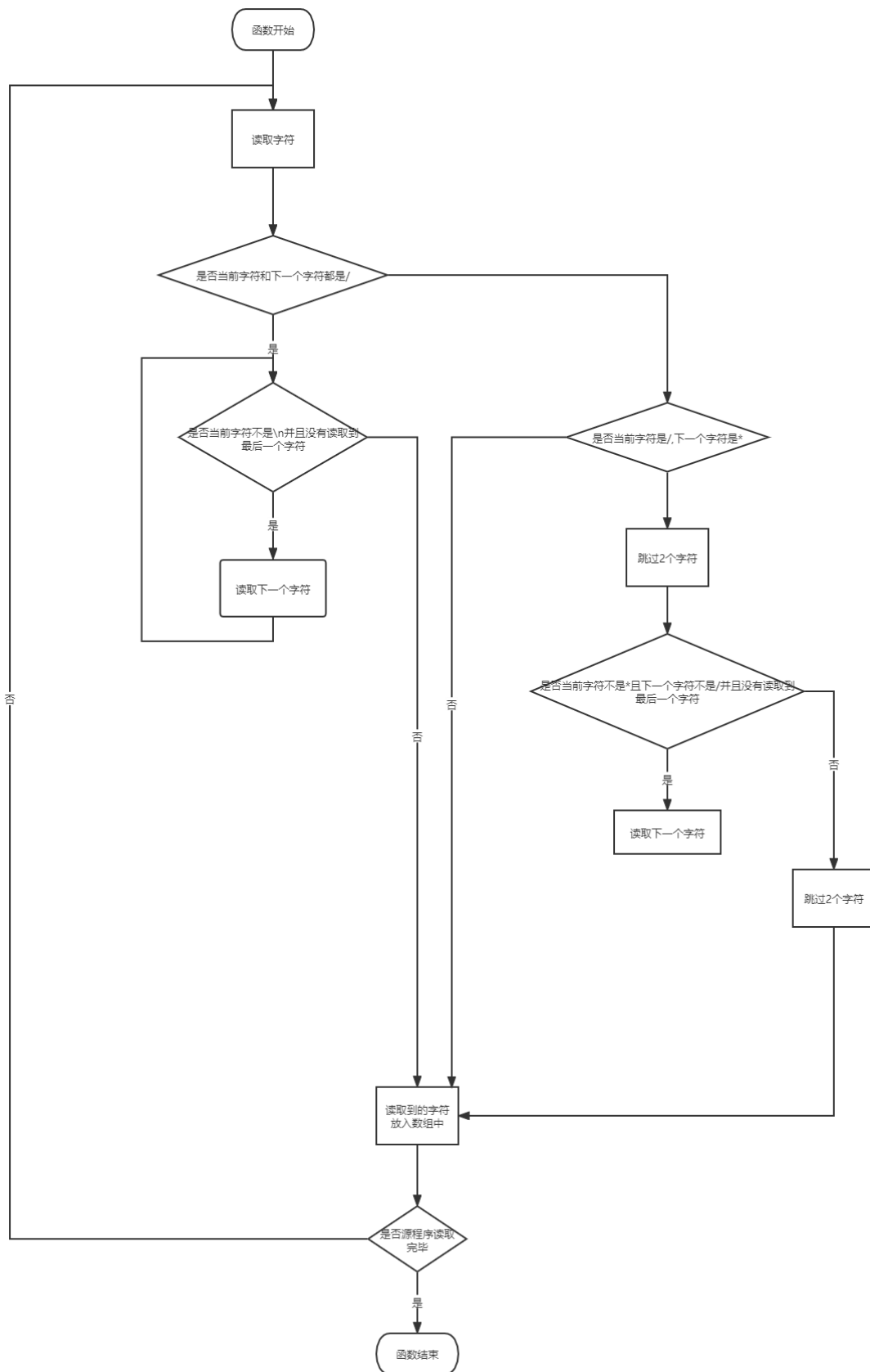
因为还处理了#include<>的情况，所以如果只有<，而没有>的话也会报错。



## 分析算法的主程序框图

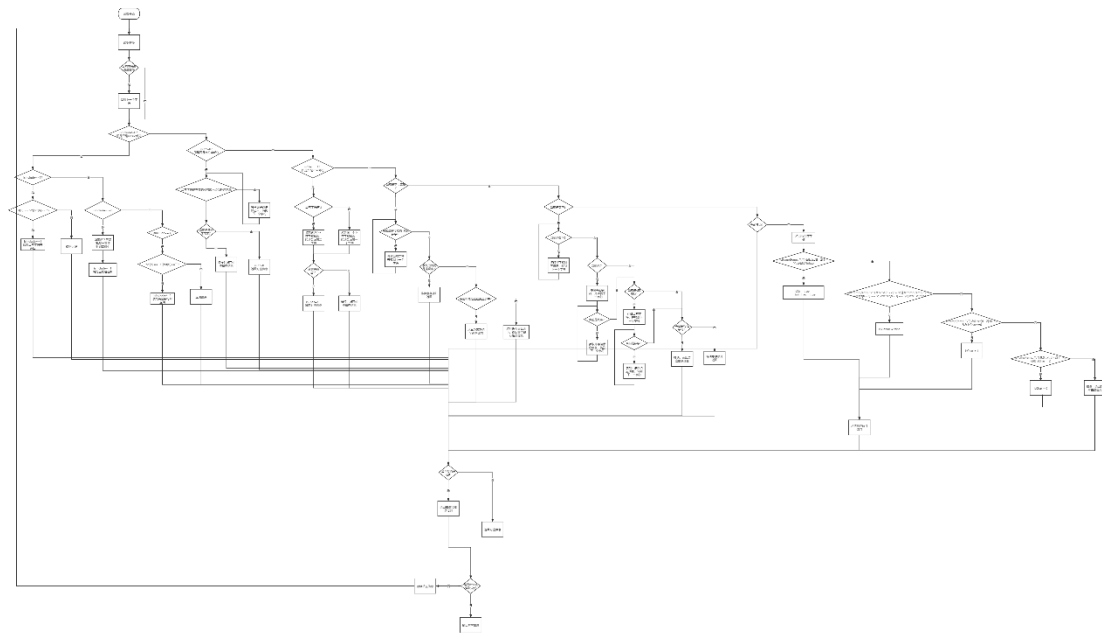


删除语句的函数流程图





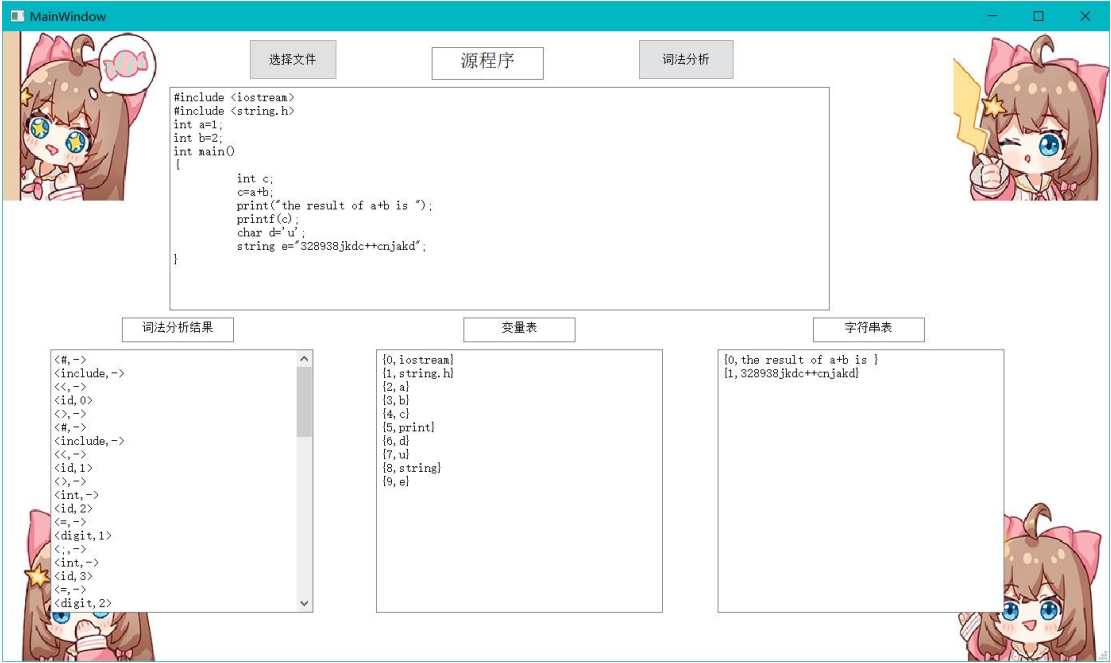
## 语法分析的函数流程图



(附件有清晰图片)

# 运行结果截图

测试代码\_1:



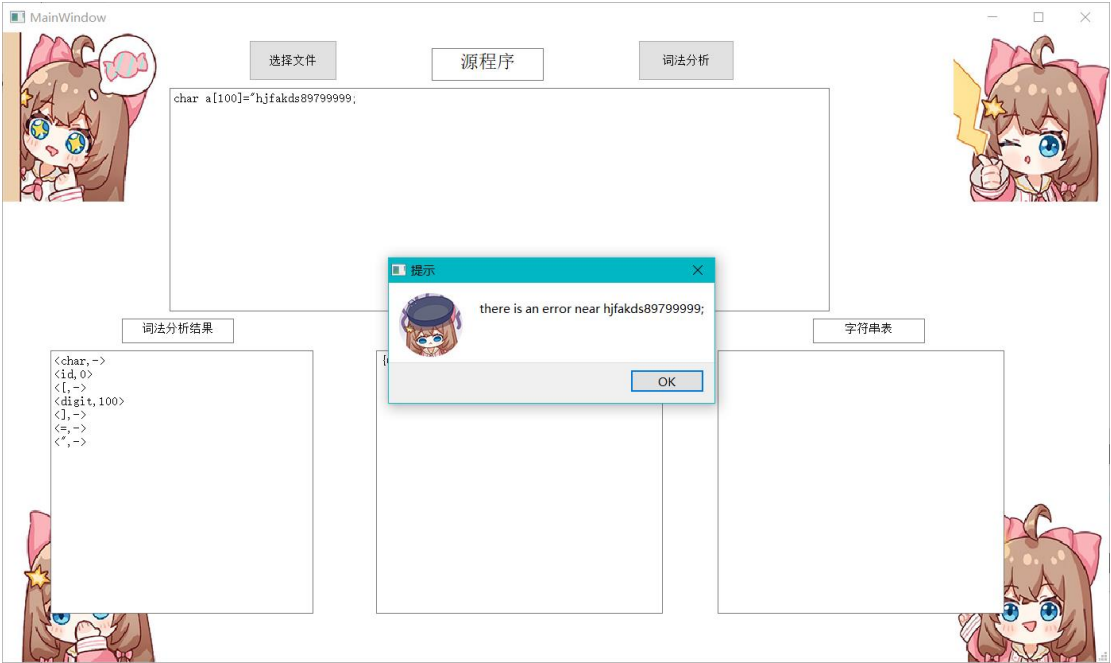
测试代码\_2:



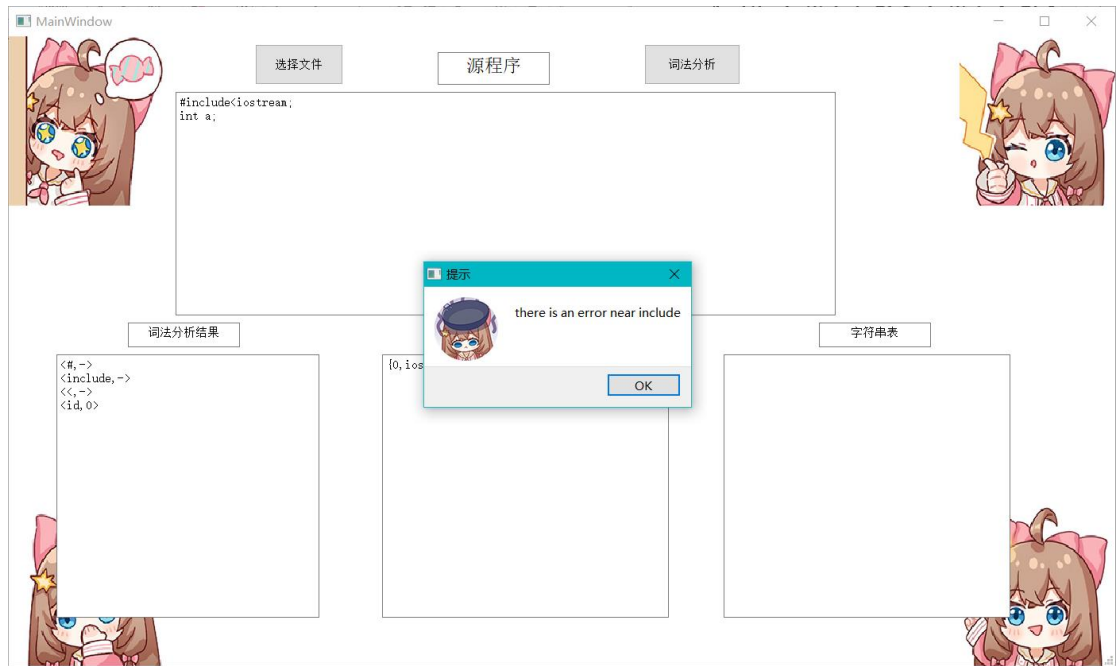
报错测试\_1:



报错测试\_2:



报错测试\_3:



#### 报错测试\_4:

