### 《编译原理》专题1\_词法分析程序设计原理与实现 实验报告

#### 22281089   陈可致

##### 实验目的

按照预先设定的正则文法集设计并实现其词法分析程序.

##### 实验内容

以下为正则文法所描述的C 语言子集单词符号的示例，请补充单词符号：

++，--， >>, <<, += , -= ,\*=, /= ，&&（逻辑与），||（逻辑或），！（逻辑非）等等，给

出补充后描述C 语言子集单词符号的正则文法，设计并实现其词法分析程序.

以下文法为左线性正则文法：

<标识符>→字母︱ <标识符>字母︱ <标识符>d

<无符号整数>→数字︱ <无符号整数>数字

<单字符分界符>→+ ︱－ ︱\* ︱；︱, ︱(︱) ︱{︱}

<双字符分界符>→<大于>=︱<小于>=︱<小于>>︱<感叹号>=︱<等于>=︱<斜竖>\*

<小于>→< <等于>→= <大于>→> <斜竖> →/

<感叹号>→!

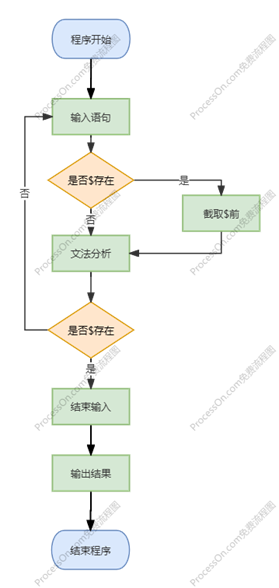
该语言的保留字 ：void、int、float、double、if、else、for、do、while 等等（也可补充）.

##### 实验环境

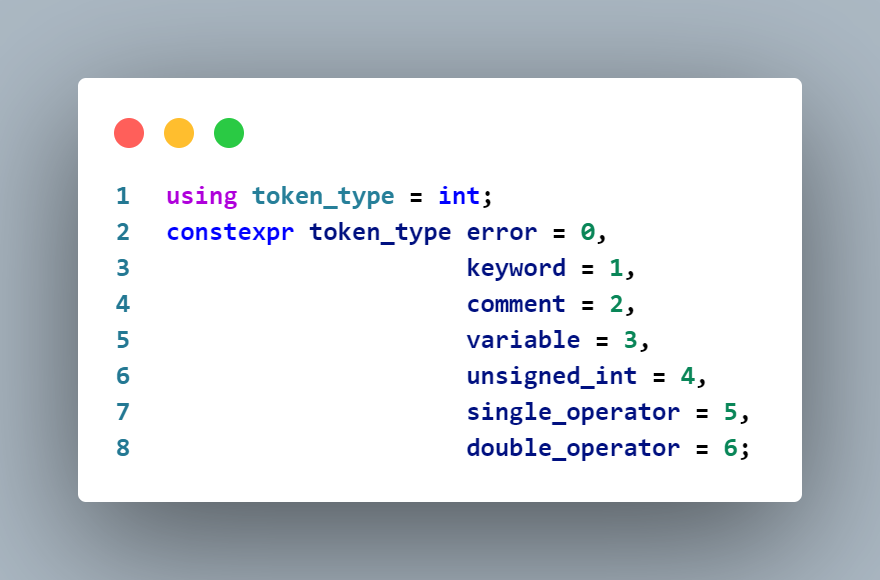
* Linux
* vscode
* gcc (Ubuntu 13.2.0-23ubuntu4) 13.2.0

##### 实验过程

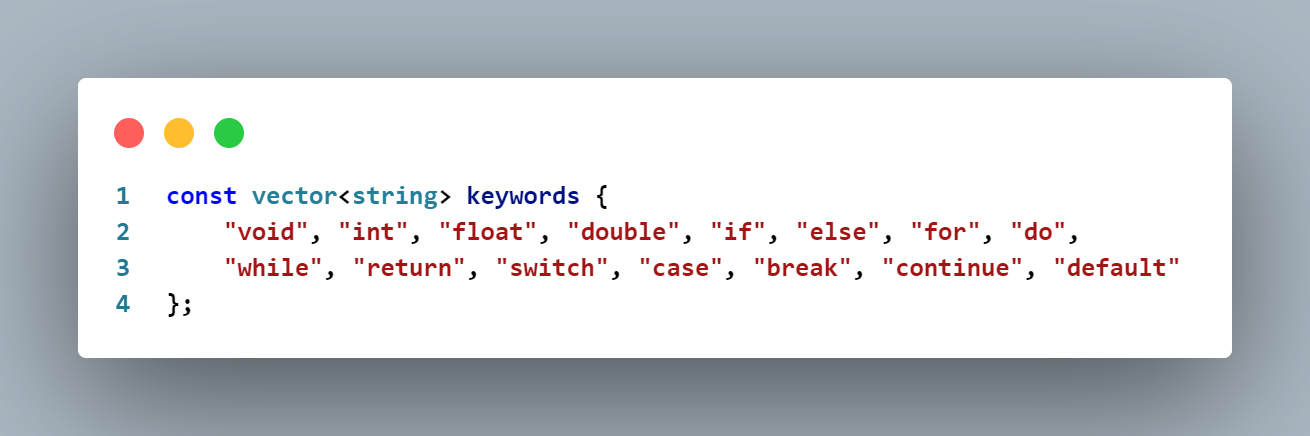
* 代码流程



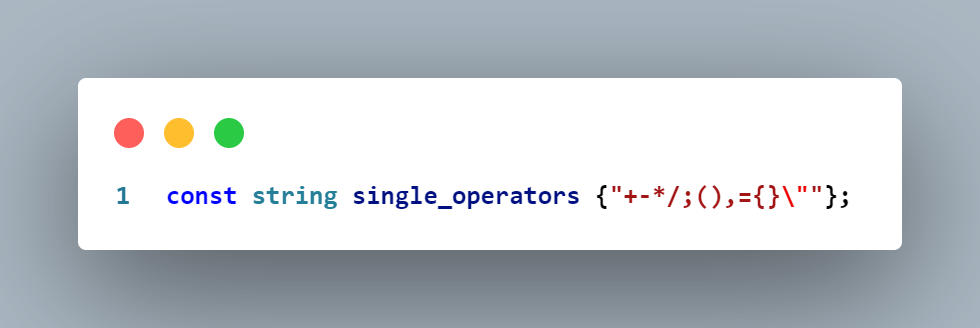
* 对类型的划分



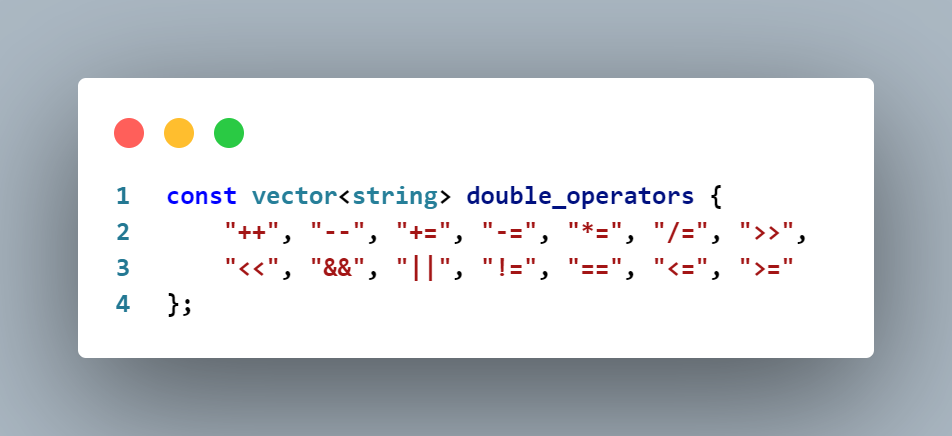
* 关键字



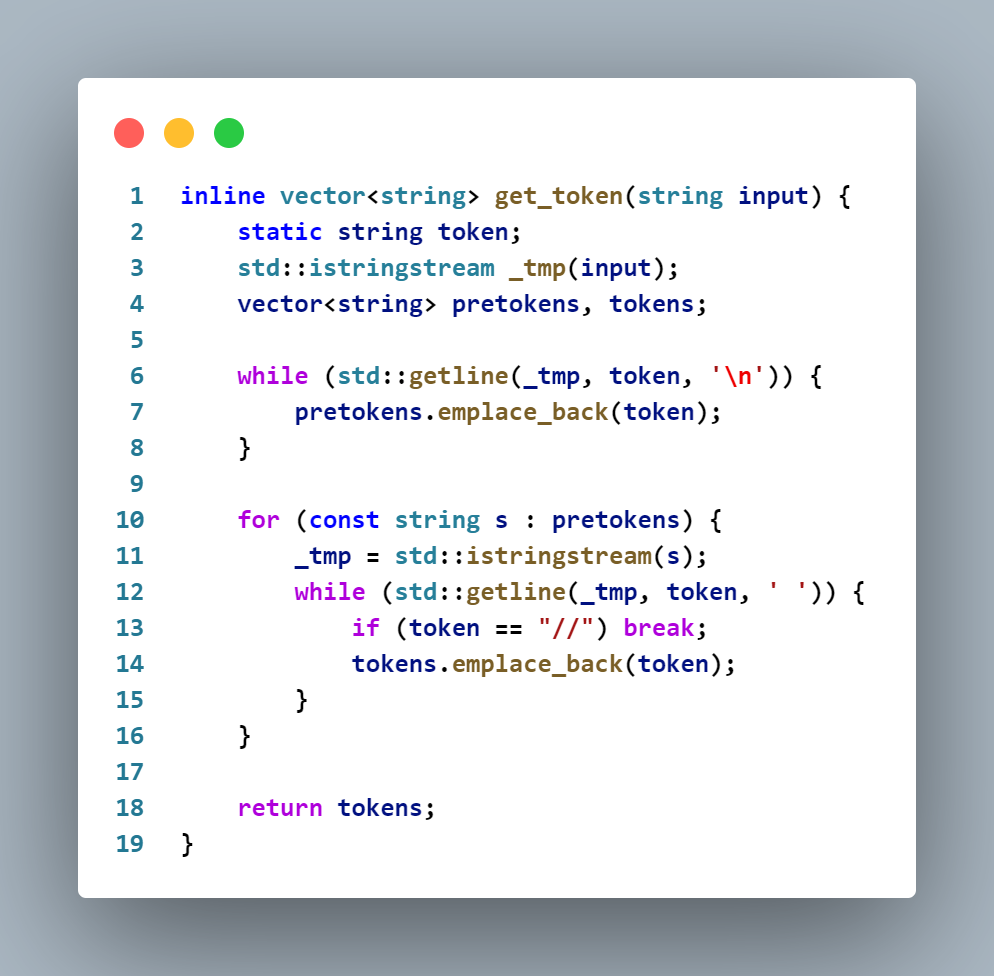
* 单字符符号



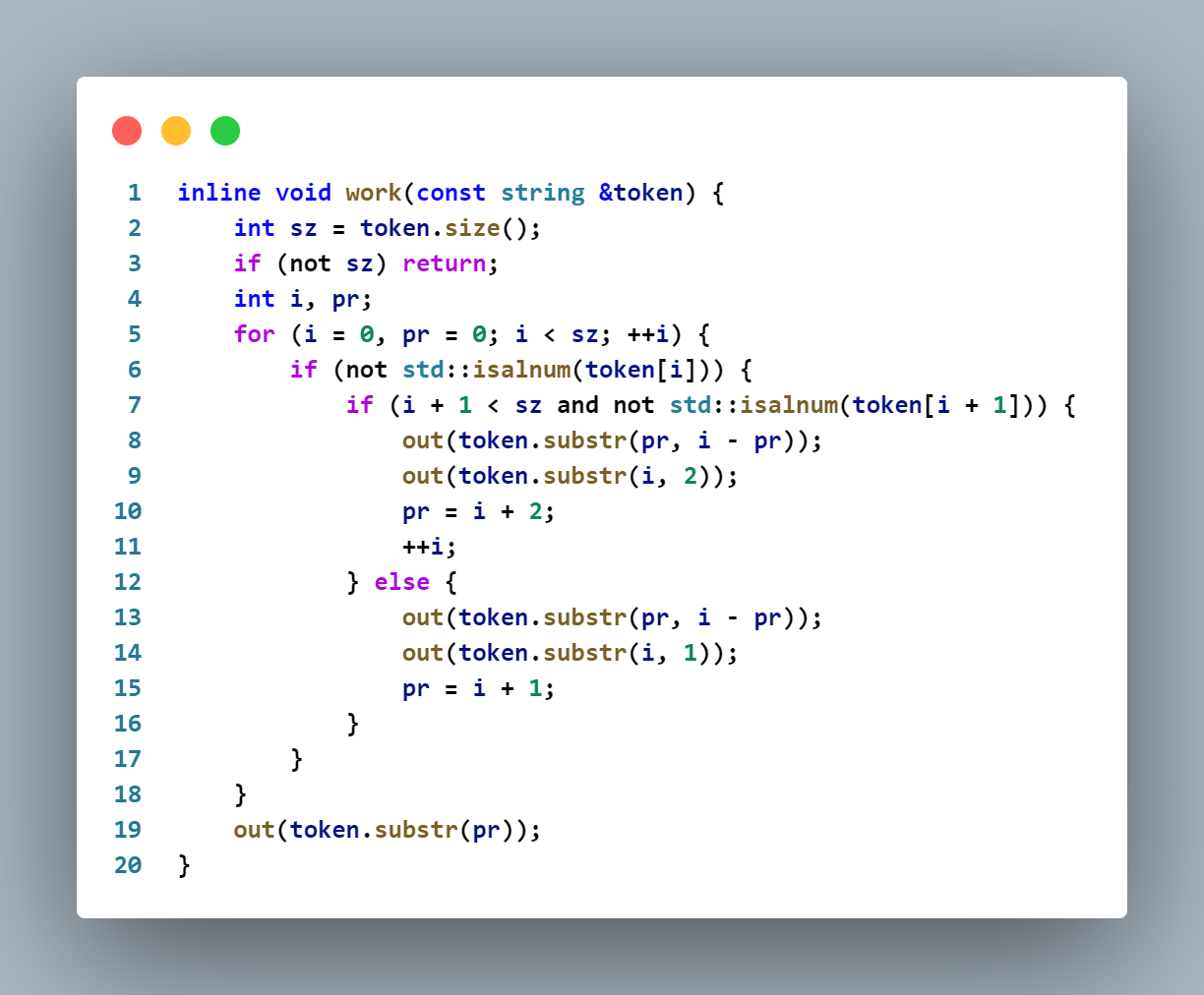
* 双字符符号



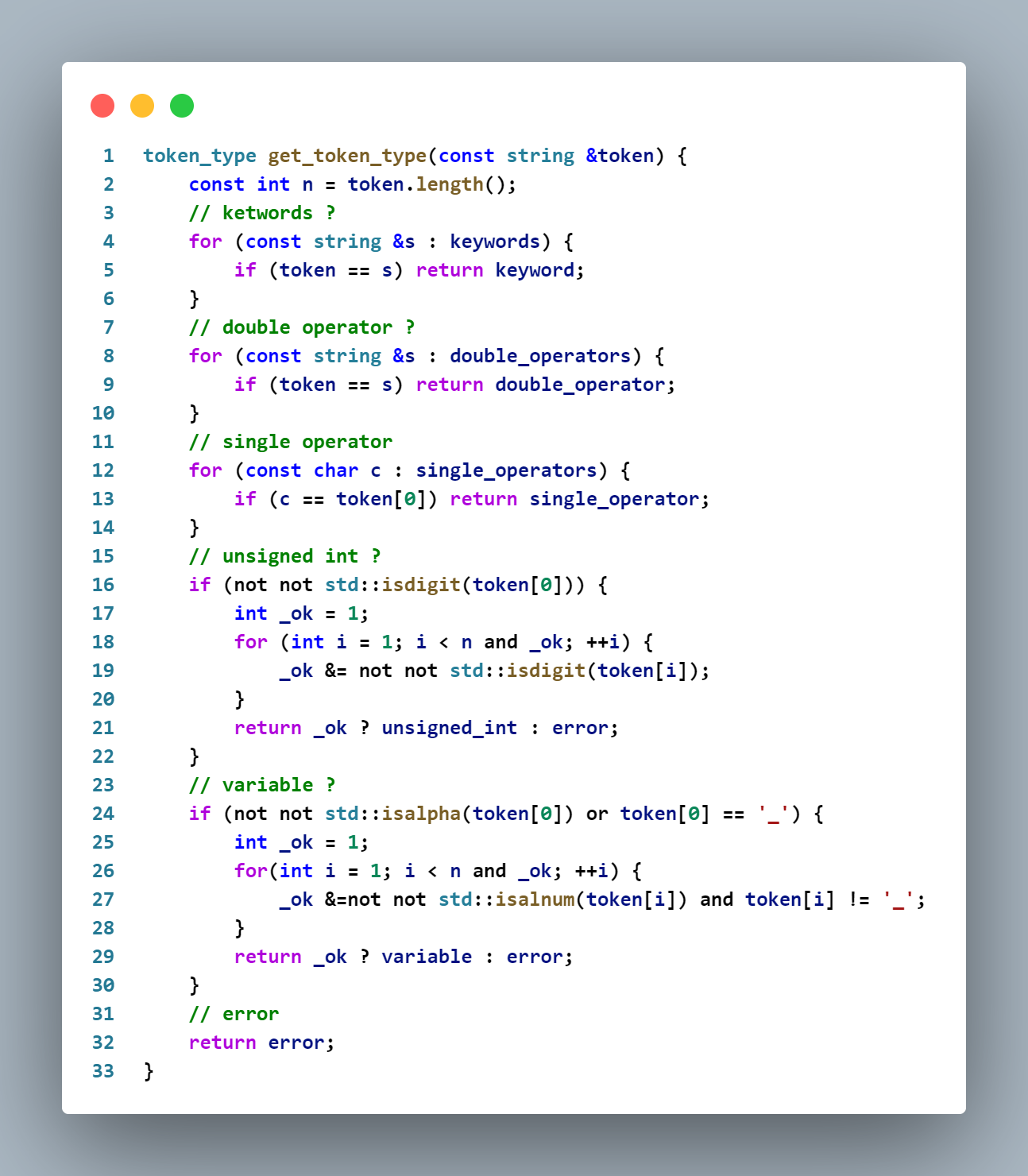
* 对token的初步划分, 利用 '\n' 和 ' ' 作初步划分



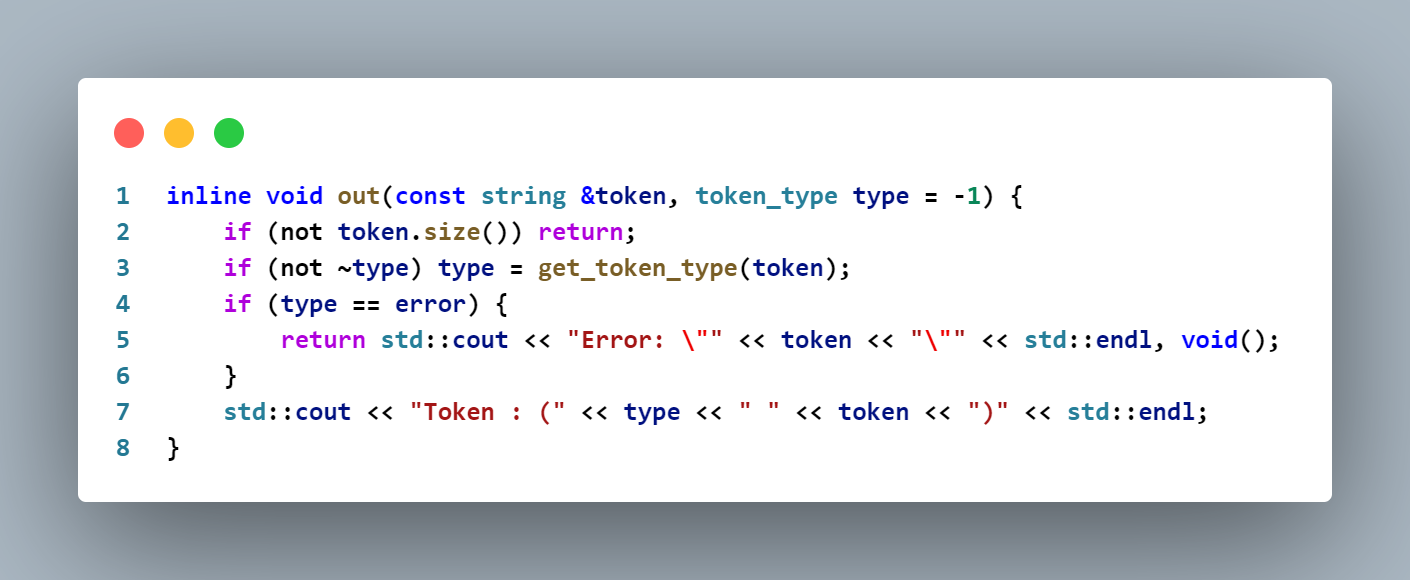
* 进一步划分token, 拆分符号与其他内容



* 对token类型的判断

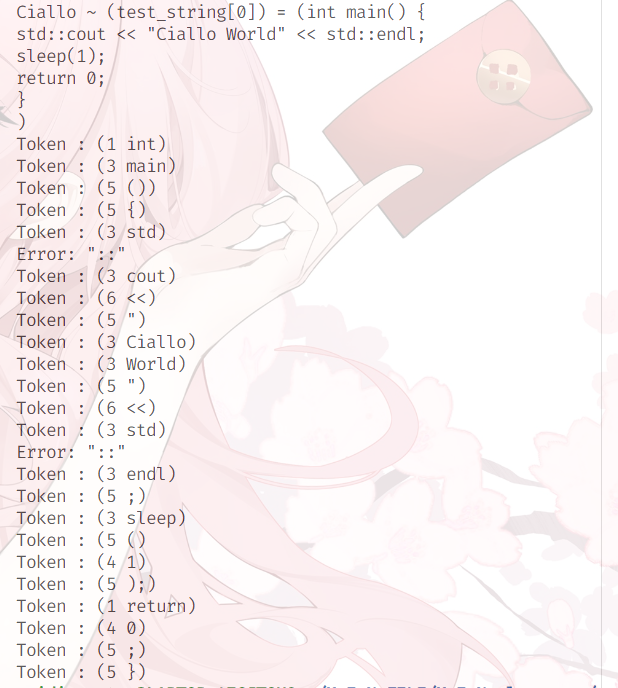


* 判断token类型并输出

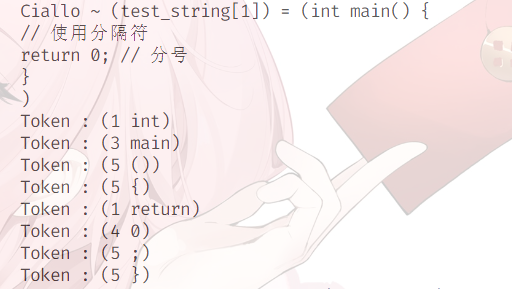


##### 实验结果

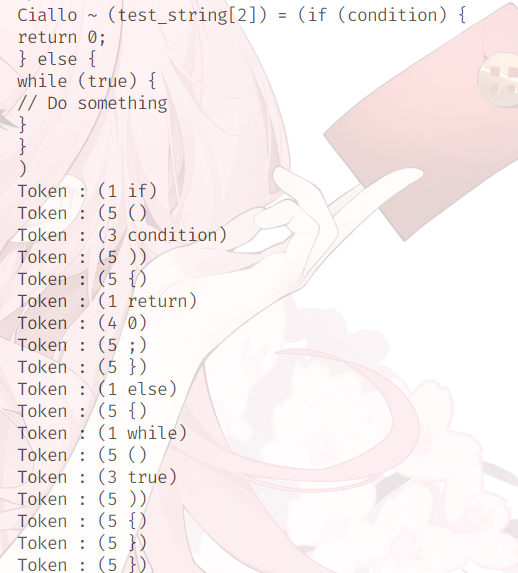
* 各项功能正常运行, 源码见附件
  + test\_0



* + test\_1



* + test\_2



* + test\_3



* 四个测试样例



##### 心得体会

有趣, 学到很多, 不仅加深了对编译原理的理解，也提升了我的编程技能. 这一经历让我认识到，编程不仅仅是解决问题，更是逻辑思维与创造力的结合. 我期待在未来的学习中，继续探索编译器的其他方面.