

编译技术课程实验报告

|  |  |
| --- | --- |
| 实验名称： | 实验一  词法分析程序编制 |
| 实验日期： | 2024-2-27 |
| 实验地点： | 西部片区4号楼208 |
|  | |
| 学号： | 33920212204567 |
| 姓名： | 任宇 |
| 专业年级： | 软工2021级 |
| 学年学期： | 2023-2024学年第二学期 |

目录

[一、 实验目的 3](#_Toc163918894)

[二、 实验内容 3](#_Toc163918895)

[三、 实验环境 3](#_Toc163918896)

[四、 实验过程 3](#_Toc163918897)

[1) 根据以下的正规式，编制正规文法，画出状态图： 3](#_Toc163918898)

[2) 程序的数据结构： 4](#_Toc163918899)

[3) 程序的算法： 4](#_Toc163918900)

[4) 测试程序： 8](#_Toc163918901)

[五、 思考题 10](#_Toc163918902)

[六、 实验心得 11](#_Toc163918903)

# 实验目的

基本掌握计算机语言的词法分析程序的开发方法。

# 实验内容

编制一个能够分析三种整数、标识符、主要运算符和主要关键字的词法分析程序。

# 实验环境

* PC微机 Windows10操作系统
* 开发环境为VS2022

# 实验过程

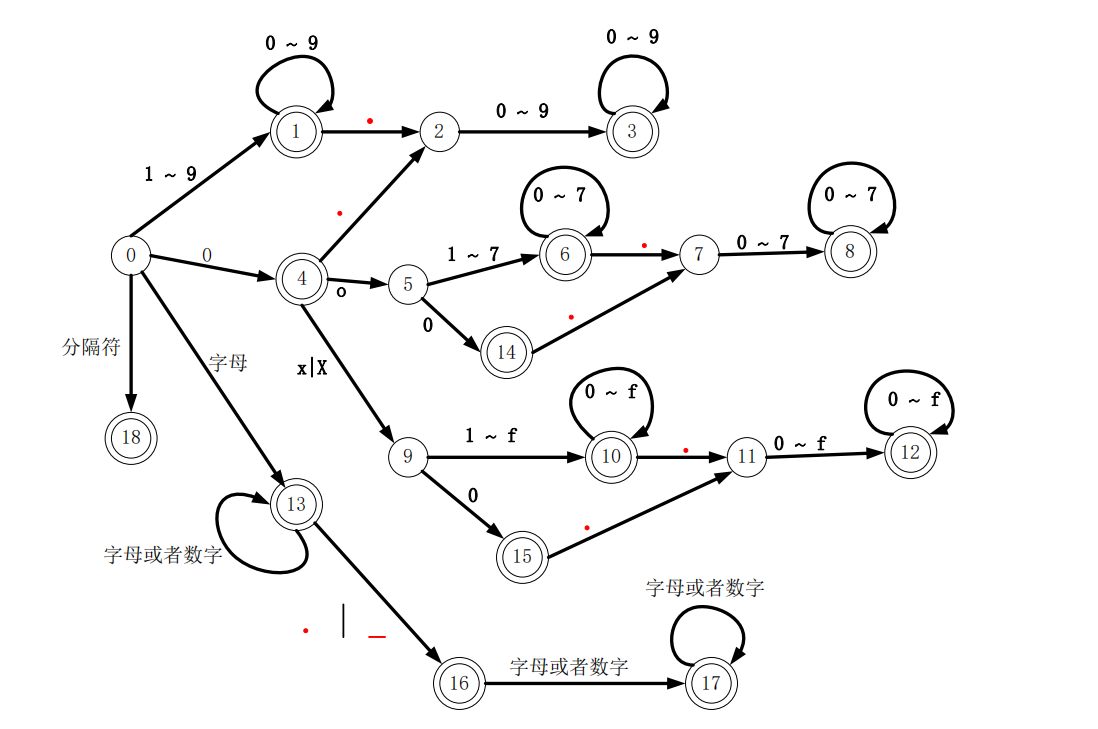
## 根据以下的正规式，编制正规文法，画出状态图：

|  |  |
| --- | --- |
| 标识符 | <字母>(<字母>|<数字字符>)\*(ε|\_)(<字母>|<数字字符>)\* |
| 十进制数 | (1|2|3|4|5|6|7|8|9)(0|1|2|3|4|5|6|7|8|9)\*|0(ε|.(0|1|2|3|4|5|6|7|8|9)(0|1|2|3|4|5|6|7|8|9)\*) |
| 八进制数 | 0o(0|1|2|3|4|5|6|7)(0|1|2|3|4|5|6|7)\*(ε|.(0|1|2|3|4|5|6|7)(0|1|2|3|4|5|6|7)\*) |
| 十六进制数 | 0x(0|1|2|3|4|5|6|7|8|9|a|b|c|d|e|f)(0|1|2|3|4|5|6|7|8|9|a|b|c|d|e|f)\*(ε|.(0|1|2|3|4|5|6|7|8|9|a|b|c|d|e|f)(0|1|2|3|4|5|6|7|8|9|a|b|c|d|e|f)\*) |
| 运算符和分隔符 | + - \* / > < = ( ) ; |
| 关键字 | if then else while do |

正规文法为：

|  |  |
| --- | --- |
| 标识符 | S → L | LT  T → L | D | \_ | TL | TD  L → a | b | ... | z | A | B | ... | Z  D → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 十进制数 | S → D | D.F  D → 0 | N  F → D | DF  N → 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1N | 2N | 3N | 4N | 5N | 6N | 7N | 8N | 9N |
| 八进制数 | S → 0o N | 0o N.F  N → D | DN  F → .D | FD  D → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 十六进制数 | S -> 0 x H A  A -> H A | ε | . B  B -> H B | ε  H → 0 | 1 | 2 | ... | 9 | a | b | ... | f | A | B | ... | F |
| 运算符和分隔符 | S -> + | - | \* | / | > | < | = | ( | ) | ; |
| 关键字 | S -> if | then | else | while | do |

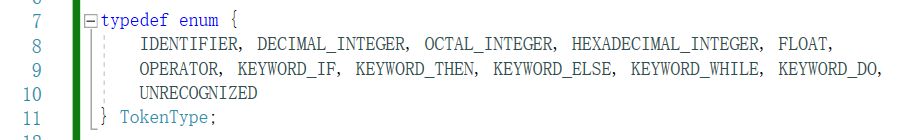
状态图为：



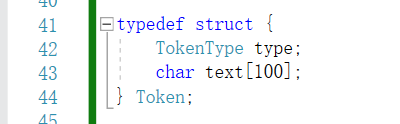
接着就可以根据状态图，设计词法分析算法。 采用C语言，设计函数scan( )实现词法分析算法并编制测试程序（主函数main）。

## 程序的数据结构：

* 枚举体TokenType：定义了一系列可能的词汇单元类型，包括标识符、整数（十进制、八进制、十六进制）、操作符、关键字（if、then、else、while、do）和未识别的类型。



* 结构体Token：用于存储单个词汇单元的类型（使用TokenType枚举）和对应的文本（字符数组）。

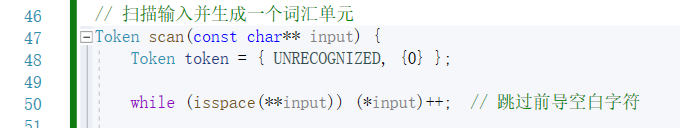


## 程序的算法：

* 本程序的核心算法流程集中在词法分析部分，即 scan 函数，其主要目的是从给定的输入字符串中识别并分类词汇单元，其流程为：

1. 跳过空白字符：

当输入的当前字符是空白（例如空格、制表符、换行符等）时，指针前移直到指向非空白字符。



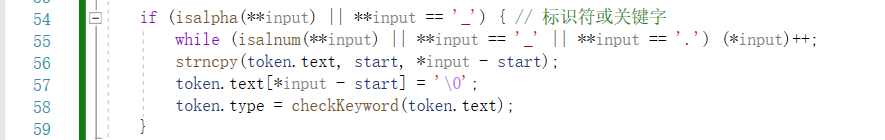
1. 开始标记词汇单元：

记录当前字符的位置，作为词汇单元开始的位置。

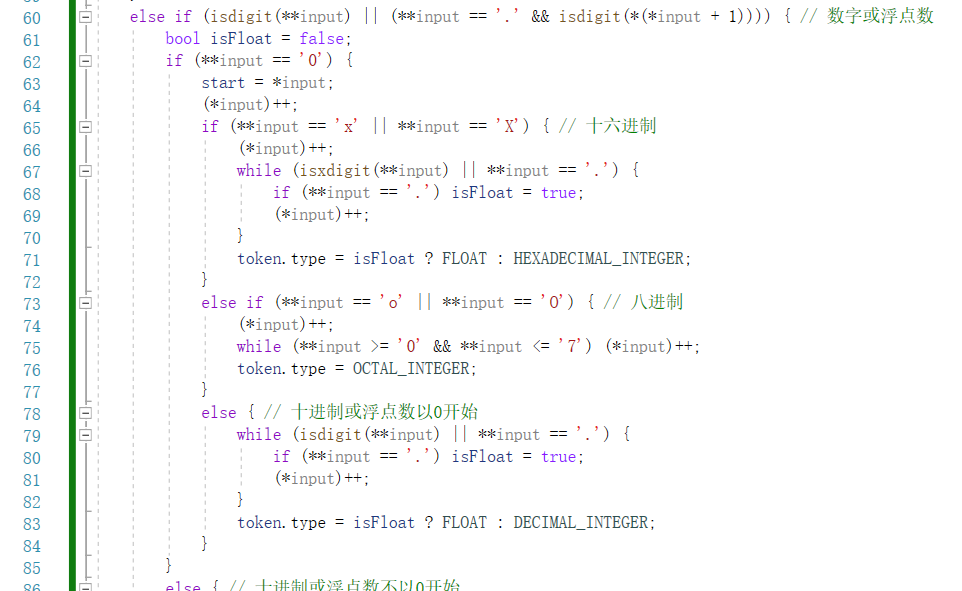


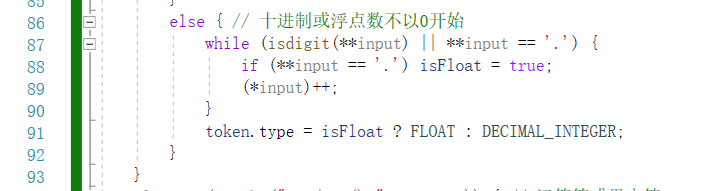
1. 分类处理：

* 标识符或关键字: 如果当前字符是字母或下划线，继续读取后续字符，直到遇到非字母、非数字、非下划线字符为止。接着使用从开始位置到当前位置的字符串去判断是否是关键字，如果不是关键字则分类为标识符。

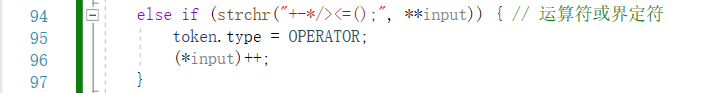


* 数字（整数或浮点数）：根据第一个字符是否为0来确定是否可能是八进制或十六进制。如果是0x或0X开头，处理为十六进制数字；如果是0o或0O开头，处理为八进制数字。对于十六进制和八进制，继续读取直到字符不再符合该进制的数字规则。对于普通的十进制数字或浮点数（以0开头或其他数字开头），读取直到遇到非数字且非点号的字符，根据是否包含点号判断是整数还是浮点数。

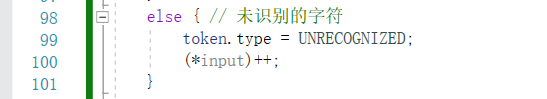




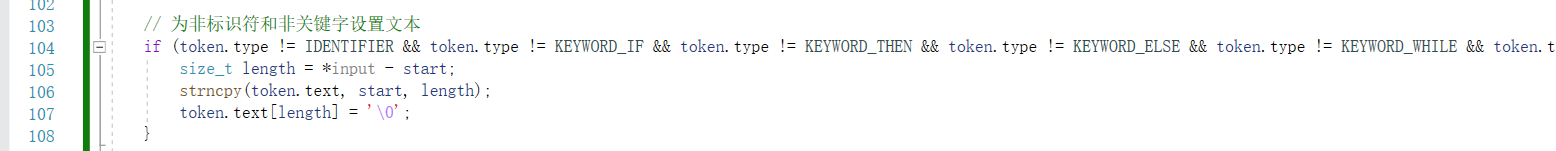
* 操作符：如果当前字符属于操作符集合，则直接分类为操作符。



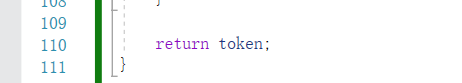
* 未识别的字符



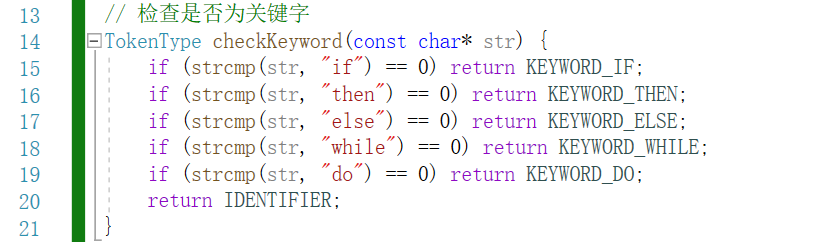
1. 设置词汇单元文本：根据已识别的词汇单元类型，将从开始位置到当前位置的文本复制到词汇单元结构的文本字段。



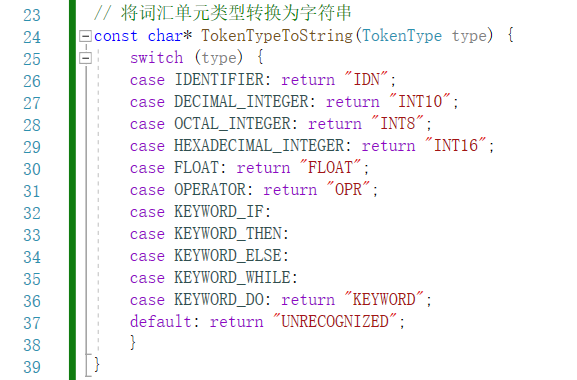
1. 返回词汇单元：



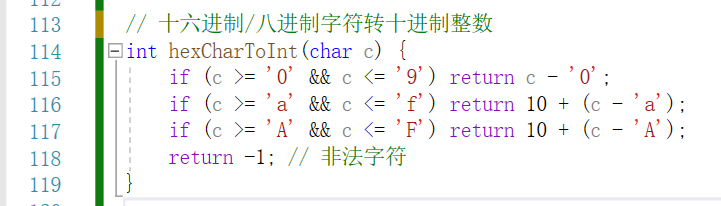
* 除此之外还有一些必要的辅助函数：
* checkKeyword：接受一个字符串，检查是否匹配某个关键字，如果是则返回相应的TokenType，否则返回IDENTIFIER。



* TokenTypeToString：将TokenType转换为对应的的字符串形式。



* hexCharToInt：将单个十六进制/八进制字符转换为其对应的整数值。



* hexStringToDouble：将十六进制或八进制字符串（可能包含小数点）转换为对应的十进制浮点数。



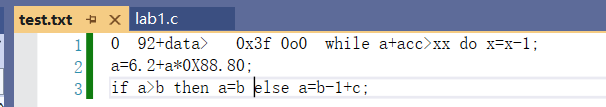
* 测评程序是否可行的主函数：

主函数 (main) 中，程序打开指定的文本文件，逐行读取内容。对每行使用 scan 函数分析，识别其中的词汇单元。对于每个词汇单元，根据其类型进行处理，如果是数字类型还可能进行数值转换，然后打印出来。文件读取完毕后关闭文件。

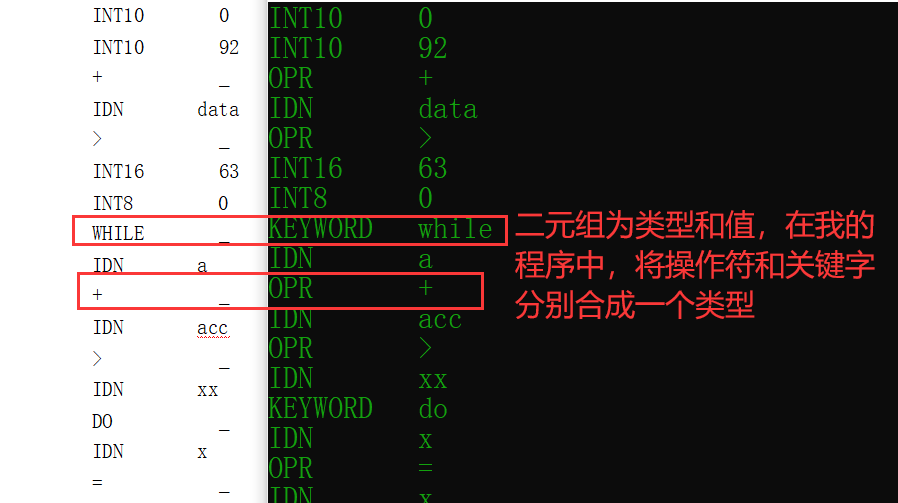


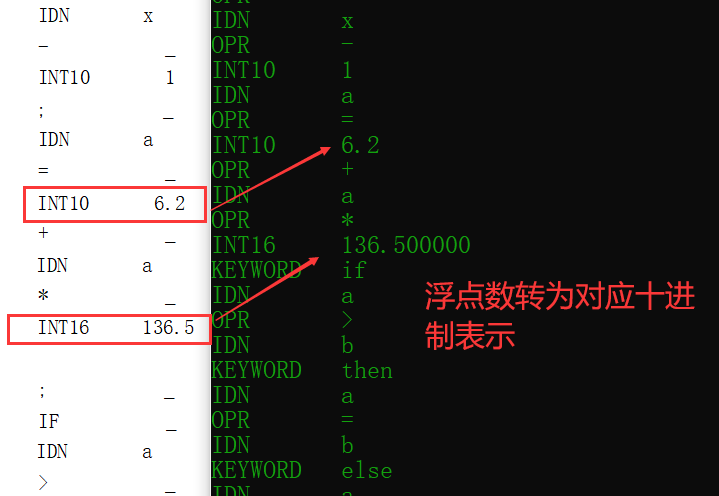
## 测试程序：

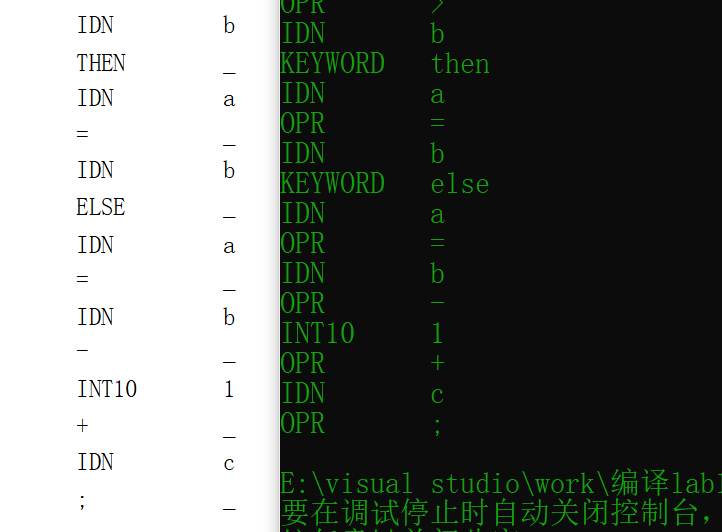
输入为：



输出为：







# 思考题

1. 词法分析能否采用空格来区分单词？

答：能否采用空格区分单词需要视情况而定，比如说在字符串或者注释中的空格不应该用作分隔符。例如，在字符串“Hello World”中，空格是字符串的一部分，不应该将其拆分为两个单独的单词，在这种情况下，词法分析器需要正确识别并处理字符串和注释。

在比如说某些不使用空格的语法结构，在某些编程语言中，如python中，空格和缩进被用来表达程序结构。同时，某些操作符或界定符可能紧挨着变量或者关键字，不由空格分隔，如a++或者！=，在这些情况下，只使用空格来进行分割是不足的。

总的来说，虽然空格是一种有效的分隔符，可以用来区分大部分单词，但词法分析器的实现还需要考虑不同的上下文和特殊情况，确保所有的词法单元都能被正确解析。

1. 程序设计中哪些环节影响词法分析的效率？如何提高效率？

答：在scan函数中，大量使用字符分类函数如isalpha,isdigit等，以及字符串操作函数如strcmp和strncpy。这些函数的频繁调用可能会导致性能下降。同时关键字的识别也很耗费时间，因为词法分析器在识别出一个标识符后，就要调用函数比较该标识符是否是关键字，若关键字表中关键字较多，则查询比较耗时长。

如果想提高效率，可以减少使用strncpy 和strcmp，尤其是在循环中。可以考虑使用指针操作来直接比较和复制字符串，或使用现代C++库如 std::string\_view来避免不必要的字符串复制。同时，可以优化检查关键字的流程，比如说有些标识符不可能是关键字，就可以不再比较。假如长度大于关键字的最大长度，就可以不进行关键字的检查。

# 实验心得

通过这次实验，我对词法分析程序的编制有了更深入的理解。实验不仅让我实际操作了从设计到实现的整个过程，还加深了我对词法分析在编译过程中的重要性的认识。在编写程序的过程中，我遇到了许多挑战，特别是在将理论转化为实际代码时。例如，处理不同类型的数值表示（十进制、八进制、十六进制）和区分关键字与普通标识符时，需要非常注意状态的转换和条件的判断。

总之，这次实验是非常宝贵的学习经验。它不仅提升了我的编程技能，也加深了我对编译原理中词法分析阶段的理解，这为完成期末大作业奠定坚实的基础。