一、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，即基本保留字、标识符、常数、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）。

二、实验预习提示

1、词法分析器的功能和输出格式

词法分析器的功能是输入源程序，输出单词符号。词法分析器的单词符号常常表示成以下的二元式(单词种别码，单词符号的属性值)。本实验中，采用的是一类符号一种别码的方式。

2、“超前搜索”方法

三、实验过程和指导

（一）准备：

1、阅读课本有关章节，明确语言的语法，写出基本保留字、标识符、常数、运算符、分隔符和程序例子。

2、初步编制好程序。

3、准备好多组测试数据。

（二）上课上机：

将源代码拷贝到机上调试，发现错误，再修改完善。

第二次上机调试通过。

（三）程序要求：

程序输入/输出示例：

如源程序为C语言。输入如下一段：

main()

{

int a,b;

a = 10;

b = a + 20;

}

要求输出如下图：

（2，”main”）

（5，”（“）

（5，”）“）

（5，”{“）

（1，”int”）

（2，”a”）

（5，”,”）

（2，”b”）

（5，”;”）

（2，”a”）

（4，”=”）

（3，”10”）

（5，”;”）

（2，”b”）

（4，”=”）

（2，”a”）

（4，”+”）

（3，”20”）

（5，”;”）

（5，”}“）

要求：

   识别保留字：if、int、for、while、do、return、break、continue；单词种别码为1。

常数为无符号整形数；单词种别码为3。

运算符包括：+、-、\*、/、=、、<、=、<=、!= ；单词种别码为4。

分隔符包括：,、;、{、}、(、)； 单词种别码为5。

其他的都识别为标识符；单词种别码为2。

（四）程序思路（仅供参考）：

这里以开始定义的C语言子集的源程序作为词法分析程序的输入数据。在词法分析中，自文件头开始扫描源程序字符，一旦发现符合“单词”定义的源程序字符串时，将它翻译成固定长度的单词内部表示，并查填适当的信息表。经过词法分析后，源程序字符串（源程序的外部表示）被翻译成具有等长信息的单词串（源程序的内部表示），并产生两个表格：常数表和标识符表，它们分别包含了源程序中的所有常数和所有标识符。

1、定义部分：定义常量、变量、数据结构。

2、初始化：从文件将源程序全部输入到字符缓冲区中。

3、取单词前：去掉多余空白。

4、取单词后：去掉多余空白（可选，看着办）。

5、取单词：读出单词的每一个字符，组成单词，分析类型。（关键是如何判断取单词结束？取到的单词是什么类型的单词？）

6、显示结果。

（五）为了能设计好程序，注意以下事情：

1、模块设计：将程序分成合理的多个模块（函数），每个模块做具体的同一事情。

2、写出（画出）设计方案：模块关系简图、流程图、全局变量、函数接口等。

3、编程时注意编程风格：空行的使用、注释的使用、缩进的使用等。