

编译原理课程设计

任务：手工编写程序实现词法分析、语法分析和语义分析

平台和语言：不限

总学时：32

时间安排：

任务	学时
词法分析	4
语法分析	12
语义分析	12
报告	4

实验一 词法分析器

要求

参考课本 3.1 和 3.2, 尤其是 3.1.2 和 3.2.4, 设计词法分析器, 识别下列单词符号:

关键字: while if else int

标识符

常数: 整常数

运算符: + - * / % = > < <= >= == != && || ! -- ++

界符: () ; [] { }

注意

- 把词法分析器安排成一个子程序, 每当语法分析器需要一个单词符号时就调用这个子程序。每次调用, 词法分析器就从输入串中识别出一个单词符号, 把它交给语法分析器。
- 单词符号的表示形式: (单词种别, 单词符号的属性值), 单词种别使用整数编码

提示

以 C 语言为例, 设表示单词符号的结构体为 TOKEN, 词法分析器为 `lexer()`, 则函数声明为:

```
TOKEN lexer();
```

由于现在没有语法分析器, 就由主函数调用 `lexer()`, 程序伪代码如下:

```
int main(){
    while(源程序未结束){
        TOKEN token=lexer();
        打印 token 的种别编码和属性值;    //若是语法分析器调用将是 return token;
    }
    return 1;
}
```

举例

输入:

```
while (i>=j) i--;
```

返回单词符号:

```
< 20,    - >
< 81,    - >
< 111,   i >
< 48,    - >
< 111,   j >
< 82,    - >
< 111,   i >
< 57,    - >
< 84,    - >
```

测试时可输出单词:

```
< while, - >
< (,     - >
< i,     i >
< >=,    - >
< j,     j >
< >,     - >
< i,     i >
< --,    - >
< ;,     - >
```

整数编码

(仅供参考):

关键字 一字一种	种别编码	属性值
int	5	-
else	15	-
if	17	-
while	20	-

标识符	种别编码	属性值
标识符	111	指针值

常数 一型一种	种别编码	属性值
整型常数	100	指针值
.....		

运算符 一符一种	种别编码	属性值
+	41	-
-	42	-
*	43	-
/	44	-
%	45	-
=	46	-
>	47	-
>=	48	-
<	49	-
<=	50	-
==	51	-
!=	52	-
&&	53	-
	54	-
!	55	-
++	56	-
--	57	-
.....		

界符 一符一种	种别编码	属性值
(81	-
)	82	-
;	84	-
{	86	-
}	87	-
[88	-
]	89	-
.....		