# 词法分析报告

## 需求分析

### 输入输出约定

输入一个完整的仅包括给定关键字及语法规则的类c程序。

输出分析后的词法及标记，格式为使用尖括号<>作为元组分割，中间先输出单词种别，然后输出单词。

### 程序功能

词法分析

## 概要设计

### 任务分解

词法分析部分可以分为两个大块，分别为数据读入输出文件调用和词法分析部分。具体设计为程序入口调用数据输入输出文件调用函数，然后调用词法分析部分进行分析输出，最后返回数据文件调用函数将词法分析结果输出为文本文件。

### 数据类型定义

```c++

struct node

{

bool isend;

int key;

int type;

char value[MAX\_SINGLE];

struct node\* nxt;

node\* next(){}

void procedure(struct node \*tmp, FILE \*file\_out){}

}

```

node为整个词法分析部分封装结构体，其中isend表示当前是否为输入程序的末尾符，key表示当前单词的单词种别，type为当前单词的单词符号的属性值，value保存当前单词，nxt为指向下一个单词的指针，next函数用于获取下一个单词信息，procedure函数将当前分析出的单词输出，并判断是否继续调用next函数进行分析。

## 详细设计

### void init()

作用：进行文件操作，打开输入文件以及输出文件。

设计：使用FILE进行文件操作，首先接收一个输入文件地址，然后会在接收到输入文件地址的相同子目录下创建或者覆盖一个output.txt存放词法分析的结果。

### node\* next()

作用：用于在词法分析器中获取下一个单词的信息。

设计：从上一次获取到的单词的下一个位置开始对源输入程序进行枚举，首先判断当前输入的是否为换行符或者空格或者tab键，若是则跳过。若输入的是完整的字母串或者字母串与数字串的结合，则和预置的关键字字串进行比较，匹配成功则将当前单词识别为关键字，结束当前函数进程；匹配失败则将当前单词识别为变量，并结束当前函数进程。如当前输入仅为数字则获得当前数字并输出，结束函数。对于其他符号则采用了一符一号的匹配方法，在这一点上显得比较麻烦，因为需要对每个符号进行单独的枚举判断，而对于比如“<=”这种两个符号组成的单一效果符还需要二次枚举判断。如果当前读入了结束符，则将isend标记标识为结束，然后返回。如果当前扫描到的字符串不满足上述情况则将其标记为词法错误。

### void procedure(struct node \*tmp, FILE \*file\_out)

作用：用于调用next()函数，并在next()返回结束符时结束函数调用。

设计：以当前next()返回状态为循环结束状态，不断地进行next()调用，直到将所有单词全部读出为止。

## 调试分析





