

```
/*C++源代码单词扫描程序（词法分析）*/
```

```
#include <iostream.h>
#include <string.h>
#include <ctype.h>
#include <stdio.h>
#include <conio.h>
```

```
char string(char c[]);           //字符串处理
函数声明
char letter(char ch);           //字母处理函
数声明
char number(char ch);          //数字处理
函数声明
char other(char ch);           //其他处理函
数声明
int search(char searchstr[],int type); //符号匹配查找函
数声明
```

```
/******全局变量的声明和定义******/
```

```
FILE *fp;                       //定义文件名，在文件流使用
char string(char c[]); //定义字符串，用于字符串处理
char ch;
```

```
//定义字符数组，用于存放关键字
```

```
char
*keyword[41]={ "include","iostream","iostream.h","string",
"stdio","stdio.h","math.h", //头文件中关键字
```

```
"int","double","float","char","long","short","enum","struct",
"typedef",//数据类型中的关键字
```

```
"if","else","switch","case","default","break","continue","re
turn", //条件选择中的关键字
```

```
"do",
"while","for","goto","Loop","cin","cout", "printf","sizeof",
//循环中的关键字
```

```
"const","static","signed","unsigned","union","volatile","voi
d","main" //其他的关键字
};
```

```
//定于字符数组，用于存放运算符
```

```
char *operate[6]={ "+","-","*","/","+","+","-"};
```

```
//定义字符数组，用于存放特殊符号
```

```
char
*symbol[31]={ "%","$","^","&","_","#","<","<=","=",">",">=",
"<>","<","=","!=","&&","||","!",
",",";", ":", "(", ")", "{",
"}","&&","||","!", "(", ">"
};
```

```
//定义字符数组，用于存放注释符号
```

```
char *note[3]={ "/*","/*","/*"};
```

```
void main () //主函数，人机交互的接口
```

```
{
    cout<<"*****C++源代码单词扫
描程序（词法分析）*****"<<endl;
    cout<<"_____
_____"<<endl;
```

```
};
    cout<<endl;
    char str,c, FileName[100];
    while(1)
    {
        cout<<"请输入需扫描分析的文件名(包括文件扩展
名):";
        cin>>FileName;
```

```
        fp=fopen(FileName,"r"); //判断文件是否
存在
```

```
        if(fp==0)
            cout<<"指定文件不存在，请输入正确的文件
名！"<<endl<<endl;
```

```
        else
        {
```

```
            cout<<endl<<FileName<<"扫描结果： "<<endl;
            cout<<"_____ "<<endl;
```

```
            str=fgetc(fp);
            while (str!=EOF)
            {
```

```
                if (isalpha(str))
                    str=letter(str);
```

```
                else
                {
```

```
                    if (isdigit(str))
                        str=number(str);
```

```
                    else
                        str=other(str);
```

```
                }
            }
        }
```

```

    }
    }
    c=getch();
}

char string(char c[])    //字符串处理函数定义
{
    int i=0 ;
    char stri[100];
    stri[i]=c[0];
    ch=fgetc(fp);
    while (ch!="")
    {
        stri[++i]=ch;
        ch=fgetc(fp);
    }
    stri[++i]=ch;
    stri[++i]='\0';
    cout<<stri<<"    "<<"字符串"<<endl;

    return (ch);
}

char letter(char ch)    //字母处理函数定义
{
    int i=-1;
    char letter[50];
    while (isalnum(ch)||ch=='.')    //是字母、头文件
处理
    {
        letter[++i]=ch;
        ch=fgetc(fp);
    }
    letter[i+1]='\0';
    if (search(letter,1))
    {
        cout<<letter<<"    "<<"关键字"<<endl;
    }
    else
    {
        cout<<letter<<"    "<<"标识符"<<endl;
    }
    return (ch);
}

```

```

char number(char ch)    //数字处理函数声明
{
    int i=-1;
    char num[20];
    while (isdigit(ch)!=0)
    {
        num[++i]=ch;
        ch=fgetc(fp);
    }
    if(isalpha(ch)!=0)
    {
        while(!isspace(ch))
        {
            num[++i]=ch;
            ch=fgetc(fp);
        }
        num[i+1]='\0';
        cout<<num<<"    "<<"error!"<<endl;
        goto loop;
    }
    num[i+1]='\0';
    cout<<num<<"    "<<"数字"<<endl;
loop: return (ch);
}

char other(char ch)    //其他函数处理定义
{
    int i=-1;
    char other[1000];
    if (isspace(ch))    //是空格
    {
        ch=fgetc(fp);
        goto loop;
    }
    while ((!isspace(ch))&&(!isalnum(ch)))    //不是空格
    且不是字母或数字
    {
        other[++i]=ch;
        ch=fgetc(fp);    //若是字母，不是空格的判断没
有写出来
    }
    other[i+1]='\0';
    if (search(other,2))
        cout<<other<<"    "<<"算数运算符"<<endl;
    else
        if (search(other,3))

```

```

        cout<<other<<"    "<<"特殊符号"<<endl;
    else
        if (search(other,3))
            cout<<other<<"    "<<" 注 释 符
"<<endl;

loop: return (ch);
}

int search(char searchstr[],int type)    //符号匹配查找
函数定义
{
    int i;
    switch (type)
    {
        case 1:
            for(i=0;i<41;i++)
            {
                if(strcmp(keyword[i],searchstr)==0)
                    return 1;
            }
        case 2:
            for(i=0;i<6;i++)
            {
                if(strcmp(operate[i],searchstr)==0)
                    return 1;
            }
            break;
        case 3:
            for(i=0;i<31;i++)
            {
                if(strcmp(symbol[i],searchstr)==0)
                    return 1;
            }
            break;
        case 4:
            for(i=0;i<3;i++)
            {
                if(!strcmp(note[i],searchstr)&& i==2)
                {
                    string(searchstr);
                    break;
                }
            }
            else
                if(!strcmp(note[i],searchstr))
                    return 1;
    }
}

```

```

break;
}
return 0;
}

```

《编译原理》试卷（A）

一、正则表达式→DFA 分析题 15 分

请写出实验一有关 C 语言词法分析中整数的正则表达式及相应的 DFA。教材 P35

二、词法分析设计题

请写出实验一有关 C 语言词法分析中整数的词法分析程序。

三、自顶向下分析设计题 15 分 教材 P136 分析表在 P116 表 4-3

请构造下面文法 G（statement）的 LL（1）分析表

statement → if-stmt | other

if-stmt → if(exp) statement else-part

else-part → else statement | ε

exp → 0 | 1

四、LR 分析题 20 分 教材 P170

请画出下面文法的 LR(1)分析表：

$S \rightarrow id \mid V := E$

$V \rightarrow id$

$E \rightarrow V \mid n$

五、语义分析题 20 分

请将下面代码段转为用四元组表达的中间代码、

if (A&&B) if (X<Y || C>D) F=1; else F=0; else if (x>) G=1;

六、综合分析设计 10 分

现有如下问题：

将一个代数的算术表达式的括号去掉。

如，表达式 a-(b-c)#，去掉括号之后的表达式为 a-b+c；

又如表达式 (a-b-c)*(c-d)#，则结果为：a*c-a*d-b*c-c*c*d。

为了处理的简单，我们做以下的规定：

（1）运算符：+、-、*、/、（）且无单目运算符

（2）操作数：小写的英文字母

（3）表达式输入的结束符为：#

请运用编译原理的所学知识解决这个问题，把解决问题的思路及概要的实现方法描述出来。

1、实验一是对 C/C++ 语言进行词法分析,请写出 C 语言注解 /* */ 的词法分析代码段.(代码段可以用 C/C++/JAVA 进行描述,也可以只画出 /* */ 的 DFA 图)

词法分析: /* */

SWITCH '/':

if(getToken=='*')

{

cout<<"/*";

l:

while(getToken!='*')

cout<<Token;

cout<<"*";

if(getToken=='/')

{

cout<<"/注释"<<endl;

}

else goto l;

}

else

2、词法分析: 对输入的类型如: 232、232.23、232E-2 的数写一个分析程序;

正则表达式:

X->AB

A->232

B->.C|E+C|E-C|空

C->2|23

递归分析程序:

void X()

{

if(token=='2')

{A();}

else error();

if(token=='.'|token=='+'|token=='-')

{ B();}

else error();

}

void A()

{

match(2);

match(3);

match(2);

}

void B()

{

if(token=='.')

{

C();

}

if(token=='E')

{

getToken();//取下一字符

if(token=='+'|token=='-')

{

C();

}

}

}

void C()

{

match(2);

getToken();

if(token=='3')

{

match(3);

}

}

void main()

{

getToken();

X();

}