# 编译原理程序设计实验报告

## ——词法分析扫描器的设计实现

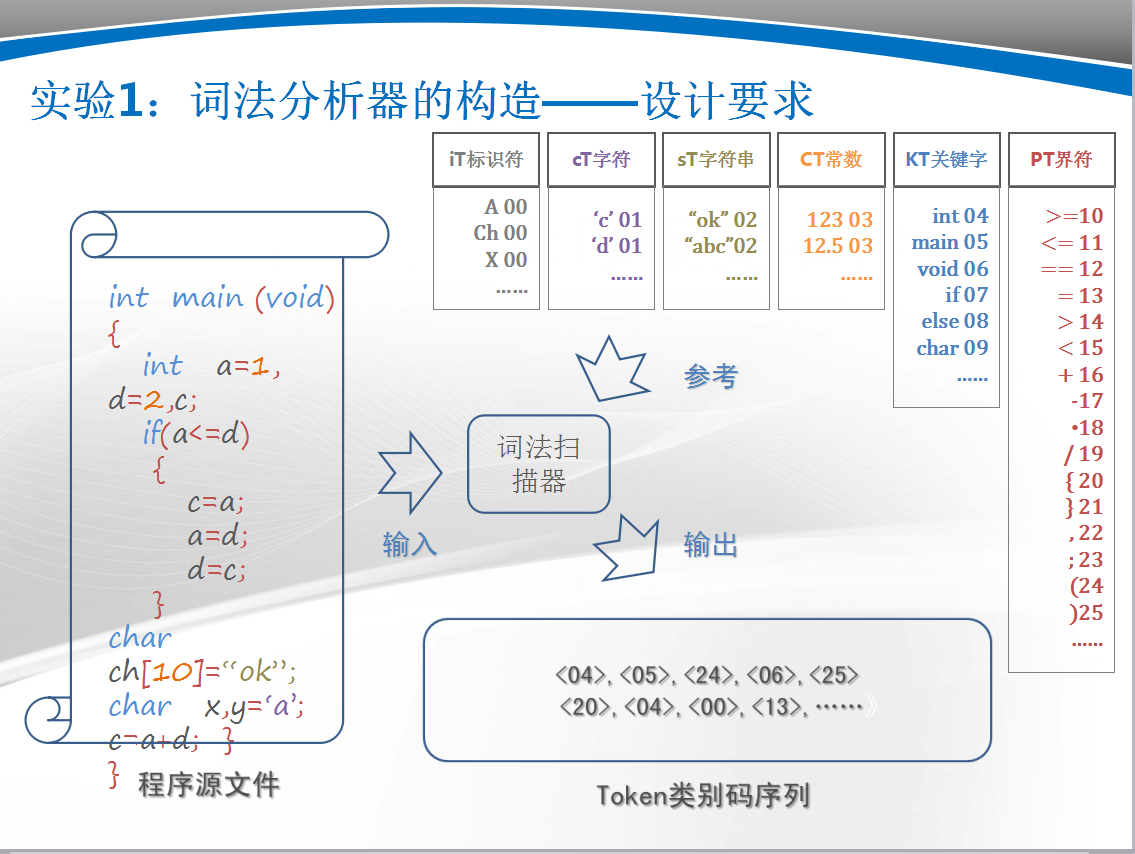
班级：计算机1605 姓名：张亚强 学号：20164382

实验目标：

在程序源文件中输入类C语言程序代码，设计词法扫描器程序，并输出。

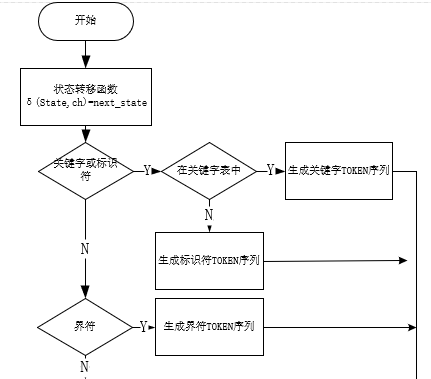
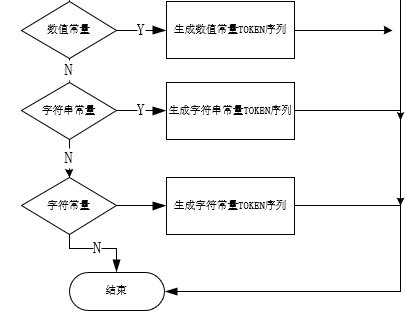
实验内容：

1. 概要设计：



2．数据结构：

6种表格均选择用vector<string>数据结构来表示，最终求得的答案选择用vector< pair<char,int> >数据结构来表示。

3.流程图：

4.关键函数：

void next\_state(char ch) //自动机状态转移函数

{

if(state == 0)

{

if(is\_Char(ch)){state=1;str.push\_back(ch);} // δ(0,字母)=1

else if(is\_Dig(ch)){state=3;str.push\_back(ch);} // δ(0,数字)=3

else if(ch == '='){state=8;str.push\_back(ch);} // δ(0,'=')=8

else if(ch == '>'){state=11;str.push\_back(ch);} // δ(0,'>')=11

else if(ch == '<'){state=14;str.push\_back(ch);} // δ(0,'<')=14

else if(ch == '+'){state=17;str.push\_back(ch);} // δ(0,'+')=17

else if(ch == '-'){state=21;str.push\_back(ch);} // δ(0,'-')=21

else if(ch == '\*'){state=25;str.push\_back(ch);} // δ(0,'\*')=25

else if(ch == '/'){state=28;str.push\_back(ch);} // δ(0,'/')=28

else if(ch == '{'){state=31;str.push\_back(ch);} // δ(0,'{')=31

else if(ch == '}'){state=32;str.push\_back(ch);} // δ(0,'}')=32

else if(ch == '('){state=33;str.push\_back(ch);} // δ(0,'(')=33

else if(ch == ')'){state=34;str.push\_back(ch);} // δ(0,')')=34

else if(ch == '['){state=35;str.push\_back(ch);} // δ(0,'[')=35

else if(ch == ']'){state=36;str.push\_back(ch);} // δ(0,']')=36

else if(ch == ','){state=37;str.push\_back(ch);} // δ(0,',')=37

else if(ch == ';'){state=38;str.push\_back(ch);} // δ(0,';')=38

else if(ch == 39){state=39;str.push\_back(ch);} // δ(0,''')=39

else if(ch == 34){state=41;str.push\_back(ch);} // δ(0,'"')=41

}

else if(state == 1)

{

if(is\_Char(ch) || is\_Dig(ch)){state=1;str.push\_back(ch);} // δ(1,字符或数字)=1

else

{

state=2;i=i-1; // δ(1,其他)=2

}

}

else if(state == 3)

{

if(is\_Dig(ch)){state=3;str.push\_back(ch);} // δ(3,数字=3

else if(ch == '.'){state=4;str.push\_back(ch);} // δ(3,'.')=4

else

{

state=7;i=i-1; // δ(3,其他)=7

}

}

else if(state == 4)

{

if(is\_Dig(ch)){state=5;str.push\_back(ch);} // δ(4,数字)=5

else

{

flag=1; //其他情况，编译出错

}

}

else if(state == 5)

{

if(is\_Dig(ch)){state=5;str.push\_back(ch);} // δ(5,数字)=5

else

{

state=6;i=i-1; // δ(5,其他)=6

}

}

else if(state == 8)

{

if(ch == '='){state=9;str.push\_back(ch);} // δ(8,'=')=9

else

{

state=10;i=i-1; // δ(8,其他)=10

}

}

else if(state == 11)

{

if(ch == '='){state=12;str.push\_back(ch);} // δ(11,'=')=12

else

{

state=13;i=i-1; // δ(11,其他)=13

}

}

else if(state == 14)

{

if(ch == '='){state=15;str.push\_back(ch);} // δ(14,'=')=15

else

{

state=16;i=i-1; // δ(14,其他)=16

}

}

else if(state ==17)

{

if(ch == '+'){state=18;str.push\_back(ch);} // δ(17,'+')=18

else if(ch == '='){state=19;str.push\_back(ch);} // δ(17,'=')=19

else

{

state=20;i=i-1; // δ(17,其他)=20

}

}

else if(state == 21)

{

if(ch == '-'){state=22;str.push\_back(ch);} // δ(21,'-')=22

else if(ch == '='){state=23;str.push\_back(ch);} // δ(21,'=')=23

else

{

state=24;i=i-1; // δ(21,其他)=24

}

}

else if(state == 25)

{

if(ch == '='){state=26;str.push\_back(ch);} // δ(25,'=')=26

else

{

state=27;i=i-1; // δ(25,其他)=27

}

}

else if(state == 28)

{

if(ch == '='){state=29;str.push\_back(ch);} // δ(28,'=')=29

else

{

state=30;i=i-1; // δ(28,其他)=30

}

}

else if(state == 39)

{

if(ch == 39){state=40;str.push\_back(ch);} // δ(39,''')=40

else

{

state=39;str.push\_back(ch); // δ(39,其他)=39

}

}

else if(state == 41)

{

if(ch == 34){state=42;str.push\_back(ch);} // δ(41,'"')=42

else

{

state=41;str.push\_back(ch); // δ(41,其他)=41

}

}

}

源程序代码：（加入注释）

#include<cstdio>

#include<vector>

#include<cstdlib>

#include<algorithm>

#include<string>

#include<iostream>

#include<utility>

#define In() freopen("in.txt","r",stdin)

#define Out() freopen("out.txt","w",stdout)

**using** **namespace** std**;**

vector**<**string**>**TOKEN\_k**;** //关键字表

vector**<**string**>**TOKEN\_p**;** //界符表

vector**<**string**>**TOKEN\_i**;** //标识符表

vector**<**string**>**TOKEN\_c**;** //数字常量表

vector**<**string**>**TOKEN\_charc**;** //字符常量表

vector**<**string**>**TOKEN\_strc**;** //字符串常量表

vector**<**pair**<**char**,**int**>** **>**ans**;** //将结果以数据对的形式保存，char表示所属表的类型，int表示在表中的位置

int state**;**

bool flag**=**0**;**

unsigned int i**=**0**;**

string str**;**

string input\_str**;**

bool is\_Char**(**char c**)** //判断是否是字符

**{**

**if(** **(**c**>=**'A' **&&** c**<=**'Z'**)** **||** **(**c**>=**'a' **&&** c**<=**'z'**)** **)return** **true;**

**else**

**return** **false;**

**}**

bool is\_Dig**(**char c**)** //判断是否是数字

**{**

**if(**c**>=**'0' **&&** c**<=**'9'**)return** **true;**

**else**

**return** **false;**

**}**

void next\_state**(**char ch**)** //自动机状态转移函数

**{**

**if(**state **==** 0**)**

**{**

**if(**is\_Char**(**ch**)){**state**=**1**;**str**.**push\_back**(**ch**);}** // δ(0,字母)=1

**else** **if(**is\_Dig**(**ch**)){**state**=**3**;**str**.**push\_back**(**ch**);}** // δ(0,数字)=3

**else** **if(**ch **==** '='**){**state**=**8**;**str**.**push\_back**(**ch**);}** // δ(0,'=')=8

**else** **if(**ch **==** '>'**){**state**=**11**;**str**.**push\_back**(**ch**);}** // δ(0,'>')=11

**else** **if(**ch **==** '<'**){**state**=**14**;**str**.**push\_back**(**ch**);}** // δ(0,'<')=14

**else** **if(**ch **==** '+'**){**state**=**17**;**str**.**push\_back**(**ch**);}** // δ(0,'+')=17

**else** **if(**ch **==** '-'**){**state**=**21**;**str**.**push\_back**(**ch**);}** // δ(0,'-')=21

**else** **if(**ch **==** '\*'**){**state**=**25**;**str**.**push\_back**(**ch**);}** // δ(0,'\*')=25

**else** **if(**ch **==** '/'**){**state**=**28**;**str**.**push\_back**(**ch**);}** // δ(0,'/')=28

**else** **if(**ch **==** '{'**){**state**=**31**;**str**.**push\_back**(**ch**);}** // δ(0,'{')=31

**else** **if(**ch **==** '}'**){**state**=**32**;**str**.**push\_back**(**ch**);}** // δ(0,'}')=32

**else** **if(**ch **==** '('**){**state**=**33**;**str**.**push\_back**(**ch**);}** // δ(0,'(')=33

**else** **if(**ch **==** ')'**){**state**=**34**;**str**.**push\_back**(**ch**);}** // δ(0,')')=34

**else** **if(**ch **==** '['**){**state**=**35**;**str**.**push\_back**(**ch**);}** // δ(0,'[')=35

**else** **if(**ch **==** ']'**){**state**=**36**;**str**.**push\_back**(**ch**);}** // δ(0,']')=36

**else** **if(**ch **==** ','**){**state**=**37**;**str**.**push\_back**(**ch**);}** // δ(0,',')=37

**else** **if(**ch **==** ';'**){**state**=**38**;**str**.**push\_back**(**ch**);}** // δ(0,';')=38

**else** **if(**ch **==** 39**){**state**=**39**;**str**.**push\_back**(**ch**);}** // δ(0,''')=39

**else** **if(**ch **==** 34**){**state**=**41**;**str**.**push\_back**(**ch**);}** // δ(0,'"')=41

**}**

**else** **if(**state **==** 1**)**

**{**

**if(**is\_Char**(**ch**)** **||** is\_Dig**(**ch**)){**state**=**1**;**str**.**push\_back**(**ch**);}** // δ(1,字符或数字)=1

**else**

**{**

state**=**2**;**i**=**i**-**1**;** // δ(1,其他)=2

**}**

**}**

**else** **if(**state **==** 3**)**

**{**

**if(**is\_Dig**(**ch**)){**state**=**3**;**str**.**push\_back**(**ch**);}** // δ(3,数字=3

**else** **if(**ch **==** '.'**){**state**=**4**;**str**.**push\_back**(**ch**);}** // δ(3,'.')=4

**else**

**{**

state**=**7**;**i**=**i**-**1**;** // δ(3,其他)=7

**}**

**}**

**else** **if(**state **==** 4**)**

**{**

**if(**is\_Dig**(**ch**)){**state**=**5**;**str**.**push\_back**(**ch**);}** // δ(4,数字)=5

**else**

**{**

flag**=**1**;** //其他情况，编译出错

**}**

**}**

**else** **if(**state **==** 5**)**

**{**

**if(**is\_Dig**(**ch**)){**state**=**5**;**str**.**push\_back**(**ch**);}** // δ(5,数字)=5

**else**

**{**

state**=**6**;**i**=**i**-**1**;** // δ(5,其他)=6

**}**

**}**

**else** **if(**state **==** 8**)**

**{**

**if(**ch **==** '='**){**state**=**9**;**str**.**push\_back**(**ch**);}** // δ(8,'=')=9

**else**

**{**

state**=**10**;**i**=**i**-**1**;** // δ(8,其他)=10

**}**

**}**

**else** **if(**state **==** 11**)**

**{**

**if(**ch **==** '='**){**state**=**12**;**str**.**push\_back**(**ch**);}** // δ(11,'=')=12

**else**

**{**

state**=**13**;**i**=**i**-**1**;** // δ(11,其他)=13

**}**

**}**

**else** **if(**state **==** 14**)**

**{**

**if(**ch **==** '='**){**state**=**15**;**str**.**push\_back**(**ch**);}** // δ(14,'=')=15

**else**

**{**

state**=**16**;**i**=**i**-**1**;** // δ(14,其他)=16

**}**

**}**

**else** **if(**state **==**17**)**

**{**

**if(**ch **==** '+'**){**state**=**18**;**str**.**push\_back**(**ch**);}** // δ(17,'+')=18

**else** **if(**ch **==** '='**){**state**=**19**;**str**.**push\_back**(**ch**);}** // δ(17,'=')=19

**else**

**{**

state**=**20**;**i**=**i**-**1**;** // δ(17,其他)=20

**}**

**}**

**else** **if(**state **==** 21**)**

**{**

**if(**ch **==** '-'**){**state**=**22**;**str**.**push\_back**(**ch**);}** // δ(21,'-')=22

**else** **if(**ch **==** '='**){**state**=**23**;**str**.**push\_back**(**ch**);}** // δ(21,'=')=23

**else**

**{**

state**=**24**;**i**=**i**-**1**;** // δ(21,其他)=24

**}**

**}**

**else** **if(**state **==** 25**)**

**{**

**if(**ch **==** '='**){**state**=**26**;**str**.**push\_back**(**ch**);}** // δ(25,'=')=26

**else**

**{**

state**=**27**;**i**=**i**-**1**;** // δ(25,其他)=27

**}**

**}**

**else** **if(**state **==** 28**)**

**{**

**if(**ch **==** '='**){**state**=**29**;**str**.**push\_back**(**ch**);}** // δ(28,'=')=29

**else**

**{**

state**=**30**;**i**=**i**-**1**;** // δ(28,其他)=30

**}**

**}**

**else** **if(**state **==** 39**)**

**{**

**if(**ch **==** 39**){**state**=**40**;**str**.**push\_back**(**ch**);}** // δ(39,''')=40

**else**

**{**

state**=**39**;**str**.**push\_back**(**ch**);** // δ(39,其他)=39

**}**

**}**

**else** **if(**state **==** 41**)**

**{**

**if(**ch **==** 34**){**state**=**42**;**str**.**push\_back**(**ch**);}** // δ(41,'"')=42

**else**

**{**

state**=**41**;**str**.**push\_back**(**ch**);** // δ(41,其他)=41

**}**

**}**

**}**

bool is\_KI**()** //关键字或标识符

**{**

**if(**state**==**2**)return** **true;**

**else**

**return** **false;**

**}**

bool is\_C**()** //数字常量

**{**

**if(**state**==**6 **||** state**==**7 **)return** **true;**

**else**

**return** **false;**

**}**

bool is\_CHARC**()** //字符常量

**{**

**if(**state **==** 40**)return** **true;**

**else**

**return** **false;**

**}**

bool is\_STR\_C**()** //字符串常量

**{**

**if(**state **==** 42**)return** **true;**

**else**

**return** **false;**

**}**

bool is\_P**()** //界符

**{**

**if(**state**==**9 **||** state**==**10 **||** state**==**12 **||** state**==**13 **||** state**==**15 **||** state**==**16 **||** **(**state**>=**18 **&&** state**<=**20**)** **||** **(**state**>=**22 **&&** state**<=**24**)** **||** state**==**26 **||** state**==**27 **||** **(**state**>=**29 **&&** state**<=**38**))return** **true;**

**else**

**return** **false;**

**}**

int main**()**

**{**

//初始化关键字表

TOKEN\_k**.**push\_back**(**"int"**);**

TOKEN\_k**.**push\_back**(**"main"**);**

TOKEN\_k**.**push\_back**(**"void"**);**

TOKEN\_k**.**push\_back**(**"if"**);**

TOKEN\_k**.**push\_back**(**"else"**);**

TOKEN\_k**.**push\_back**(**"char"**);**

TOKEN\_k**.**push\_back**(**"float"**);**

TOKEN\_k**.**push\_back**(**"double"**);**

TOKEN\_k**.**push\_back**(**"for"**);**

TOKEN\_k**.**push\_back**(**"while"**);**

TOKEN\_k**.**push\_back**(**"break"**);**

TOKEN\_k**.**push\_back**(**"switch"**);**

TOKEN\_k**.**push\_back**(**"case"**);**

TOKEN\_k**.**push\_back**(**"default"**);**

TOKEN\_k**.**push\_back**(**"return"**);**

//初始化界符表

TOKEN\_p**.**push\_back**(**"<="**);** //0

TOKEN\_p**.**push\_back**(**">="**);** //1

TOKEN\_p**.**push\_back**(**"=="**);** //2

TOKEN\_p**.**push\_back**(**"="**);** //3

TOKEN\_p**.**push\_back**(**">"**);** //4

TOKEN\_p**.**push\_back**(**"<"**);** //5

TOKEN\_p**.**push\_back**(**"+"**);** //6

TOKEN\_p**.**push\_back**(**"++"**);** //7

TOKEN\_p**.**push\_back**(**"+="**);** //8

TOKEN\_p**.**push\_back**(**"-"**);** //9

TOKEN\_p**.**push\_back**(**"--"**);** //10

TOKEN\_p**.**push\_back**(**"-="**);** //11

TOKEN\_p**.**push\_back**(**"\*"**);** //12

TOKEN\_p**.**push\_back**(**"\*="**);** //13

TOKEN\_p**.**push\_back**(**"/"**);** //14

TOKEN\_p**.**push\_back**(**"/="**);** //15

TOKEN\_p**.**push\_back**(**"{"**);** //16

TOKEN\_p**.**push\_back**(**"}"**);** //17

TOKEN\_p**.**push\_back**(**"("**);** //18

TOKEN\_p**.**push\_back**(**")"**);** //19

TOKEN\_p**.**push\_back**(**"["**);** //20

TOKEN\_p**.**push\_back**(**"]"**);** //21

TOKEN\_p**.**push\_back**(**","**);** //22

TOKEN\_p**.**push\_back**(**";"**);** //23

In**();**Out**();** //从in.txt文件中读入数据，并将数据结果写入out.txt中

state**=**0**;**flag**=**0**;**

str**.**clear**();**

**while(**getline**(**cin**,**input\_str**))**

**{**

**if(**input\_str **==** "#"**)break;** //文件以'#'结尾

i**=**0**;**

**while(**i**<**input\_str**.**size**())**

**{**

flag**=**0**;**

//printf("state = %d \n",state);

next\_state**(**input\_str**[**i**]);** //调用状态转移函数

**if(**flag **==**1**){break;}** //编译出错

**if(**is\_KI**())** //关键字或标识符

**{**

bool flag\_k**=**0**,**flag\_i**=**0**;**

//在key表中查询

**for(**unsigned int j**=**0**;**j**<**TOKEN\_k**.**size**();**j**++)**

**{**

**if(**str **=**TOKEN\_k**[**j**]){**

ans**.**push\_back**(**make\_pair**(**'k'**,**j**));** flag\_k**=**1**;}**

**}**

**if(**flag\_k **==** 0**)** //key表中无此串

**{**

//在标识符表中查询

**for(**unsigned int j**=**0**;**j**<**TOKEN\_i**.**size**();**j**++)**

**{**

**if(**str**==**TOKEN\_i**[**j**]){**

ans**.**push\_back**(**make\_pair**(**'i'**,**j**));** flag\_i**=**1**;}**

**}**

**if(**flag\_i **==** 0**){**

TOKEN\_i**.**push\_back**(**str**);**ans**.**push\_back**(**make\_pair**(**'i'**,**TOKEN\_i**.**size**()-**1**));}**

**}** //表中查不到，将其加到表尾

state**=**0**;**

str**.**clear**();**

**}**

**else** **if(**is\_P**())** //界符

**{**

bool flag\_p**=**0**;**

**for(**unsigned int j**=**0**;**j**<**TOKEN\_p**.**size**();**j**++)** //在界符表中查询

**{**

**if(**str **==** TOKEN\_p**[**j**]){**

ans**.**push\_back**(**make\_pair**(**'p'**,**j**));** flag\_p**=**1**;}**

**}**

**if(**flag\_p **==** 0**){**

TOKEN\_p**.**push\_back**(**str**);**ans**.**push\_back**(**make\_pair**(**'p'**,**TOKEN\_p**.**size**()-**1**));}**

//表中查不到，将其加到表尾

state**=**0**;**

str**.**clear**();**

**}**

**else** **if(**is\_C**())** //数值常量

**{**

bool flag\_c**=**0**;**

//在常量表中查询

**for(**unsigned int j**=**0**;**j**<**TOKEN\_c**.**size**();**j**++)**

**{**

**if(**str **==** TOKEN\_c**[**j**]){**

ans**.**push\_back**(**make\_pair**(**'c'**,**j**));** flag\_c**=**1**;}**

**}**

**if(**flag\_c **==** 0**){**

TOKEN\_c**.**push\_back**(**str**);**ans**.**push\_back**(**make\_pair**(**'c'**,**TOKEN\_c**.**size**()-**1**));}**

//表中查不到，将其加到表尾

state**=**0**;**

str**.**clear**();**

**}**

**else** **if(**is\_STR\_C**())** //字符串常量

**{**

bool flag\_strc**=**0**;**

//在常量表中查询

**for(**unsigned int j**=**0**;**j**<**TOKEN\_strc**.**size**();**j**++)**

**{**

**if(**str **==** TOKEN\_strc**[**j**]){**

ans**.**push\_back**(**make\_pair**(**'s'**,**j**));** flag\_strc**=**1**;}**

**}**

**if(**flag\_strc **==** 0**){**

TOKEN\_strc**.**push\_back**(**str**);**ans**.**push\_back**(**make\_pair**(**'s'**,**TOKEN\_strc**.**size**()-**1**));}**

//表中查不到，将其加到表尾

state**=**0**;**

str**.**clear**();**

**}**

**else** **if(**is\_CHARC**())** //字符常量

**{**

bool flag\_charc**=**0**;**

//在常量表中查询

**for(**unsigned int j**=**0**;**j**<**TOKEN\_charc**.**size**();**j**++)**

**{**

**if(**str **==** TOKEN\_charc**[**j**]){**

ans**.**push\_back**(**make\_pair**(**'C'**,**j**));** flag\_charc**=**1**;}**

**}**

**if(**flag\_charc **==** 0**){**

TOKEN\_charc**.**push\_back**(**str**);**ans**.**push\_back**(**make\_pair**(**'C'**,**TOKEN\_charc**.**size**()-**1**));}**

//表中查不到，将其加到表尾

state**=**0**;**

str**.**clear**();**

**}**

i**++;**

**}**

**if(**flag**==**1**){**printf**(**"该程序编译出错!!\n"**);break;}**

**}**

**for(**unsigned int j**=**0**;**j**<**ans**.**size**();**j**++)** //输出结果

**{**

printf**(**"( %c , %d )\n"**,**ans**[**j**].**first**,**ans**[**j**].**second**);**

**if(**ans**[**j**].**first **==**'k'**)**

**{**

cout**<<** TOKEN\_k**[**ans**[**j**].**second**]** **<<**endl**;**

**}**

**else** **if(**ans**[**j**].**first **==**'i'**)**

**{**

cout**<<** TOKEN\_i**[**ans**[**j**].**second**]** **<<**endl**;**

**}**

**else** **if(**ans**[**j**].**first **==**'p'**)**

**{**

cout**<<** TOKEN\_p**[**ans**[**j**].**second**]** **<<**endl**;**

**}**

**else** **if(**ans**[**j**].**first **==**'c'**)**

**{**

cout**<<** TOKEN\_c**[**ans**[**j**].**second**]** **<<**endl**;**

**}**

**else** **if(**ans**[**j**].**first **==**'s'**)**

**{**

cout**<<** TOKEN\_strc**[**ans**[**j**].**second**]** **<<**endl**;**

**}**

**else** **if(**ans**[**j**].**first **==**'C'**)**

**{**

cout**<<** TOKEN\_charc**[**ans**[**j**].**second**]** **<<**endl**;**

**}**

**}**

**return** 0**;**

**}**

/\*

int main(void)

{

int a=1,d=2,c;

float pi =3.14159;

if(a<=d)

{

c=a;

a=d;

d=c;

}

char cA[10]="ok";

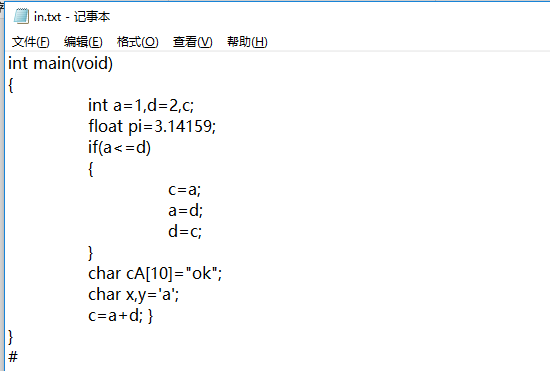
char x,y='a';

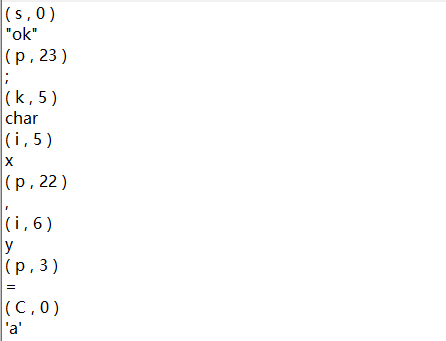
c=a+d; }

}

\*/

程序运行结果：（截屏）

输入：

输出：