

**编译原理课程设计**

实验报告

北京航空航天大学

计算机学院

陈麒先

16061160

二○一八年十一月

**郑重声明**

**关于诚实守信公约：**

**本实验报告中参考了文献或互联网资料的部分均有引用标注，另有与老师或同学研究后的成果引用将在特别致谢中说明，其余部分均为本人通过查阅课件、课堂笔记和教材后独立思考的结果。另，实验报告撰写仓促，如有错误，在所难免，欢迎批评指正！特此声明。**

**陈麒先**

**原创性声明**

**本实验报告未注释部分内容由作者原创。抄袭行为在任何情况下都是不能容忍的(COPY is strictly prohibited under any circumstances)！转载或引用须征得作者本人同意，并注明出处！勿谓言之不预！**

**陈麒先**



# 词法分析测试用例

同上次作业中的test。

# 词法分析结果

CONST const

INT int

IDEN a

EQUAL =

PLUS +

UINT 1

SEMI ;

CONST const

INT int

IDEN B

EQUAL =

UINT 0

COMMA ,

IDEN \_

EQUAL =

MINUS -

UINT 30

COMMA ,

IDEN \_d

EQUAL =

UINT 49

SEMI ;

CONST const

CHAR char

IDEN c\_

EQUAL =

CHAR 'c'

SEMI ;

CONST const

CHAR char

IDEN c\_0

EQUAL =

CHAR '\_'

COMMA ,

IDEN c1

EQUAL =

CHAR '+'

COMMA ,

IDEN \_c\_2

EQUAL =

CHAR '-'

COMMA ,

IDEN C3

EQUAL =

CHAR '\*'

COMMA ,

IDEN C\_4

EQUAL =

CHAR '/'

COMMA ,

IDEN \_C\_5\_

EQUAL =

CHAR '0'

SEMI ;

INT int

IDEN num

SEMI ;

CHAR char

IDEN ch

SEMI ;

INT int

IDEN array

LZKH [

UINT 100

RZKH ]

SEMI ;

CHAR char

IDEN str

LZKH [

UINT 10

RZKH ]

SEMI ;

INT int

IDEN n1

COMMA ,

IDEN n\_2

COMMA ,

IDEN N3

COMMA ,

IDEN array1

LZKH [

UINT 10

RZKH ]

COMMA ,

IDEN N\_4

SEMI ;

CHAR char

IDEN s\_0

LZKH [

UINT 9

RZKH ]

COMMA ,

IDEN ch1

COMMA ,

IDEN C2

COMMA ,

IDEN c\_3

COMMA ,

IDEN Ch\_4

SEMI ;

INT int

IDEN add

LPAR (

INT int

IDEN x

COMMA ,

INT int

IDEN y

RPAR )

LDKH {

RETURN return

LPAR (

IDEN x

PLUS +

IDEN y

RPAR )

SEMI ;

RDKH }

CHAR char

IDEN getch

LPAR (

RPAR )

LDKH {

CONST const

CHAR char

IDEN ch

EQUAL =

CHAR '\_'

SEMI ;

RETURN return

LPAR (

IDEN ch

RPAR )

SEMI ;

RDKH }

INT int

IDEN fac

LPAR (

INT int

IDEN n

RPAR )

LDKH {

IF if

LPAR (

IDEN n

ASSIGN ==

UINT 1

RPAR )

RETURN return

LPAR (

UINT 1

RPAR )

SEMI ;

RETURN return

LPAR (

IDEN n

MULT \*

IDEN fac

LPAR (

IDEN n

MINUS -

UINT 1

RPAR )

RPAR )

SEMI ;

RDKH }

VOID void

IDEN print\_hello

LPAR (

RPAR )

LDKH {

PRINTF printf

LPAR (

STRING "Hello world!98"

RPAR )

SEMI ;

RDKH }

VOID void

IDEN print\_num

LPAR (

INT int

IDEN n

RPAR )

LDKH {

PRINTF printf

LPAR (

IDEN n

RPAR )

SEMI ;

RDKH }

VOID void

IDEN print\_char

LPAR (

CHAR char

IDEN c

RPAR )

LDKH {

PRINTF printf

LPAR (

IDEN c

RPAR )

SEMI ;

RDKH }

VOID void

IDEN print\_all

LPAR (

INT int

IDEN a

RPAR )

LDKH {

PRINTF printf

LPAR (

STRING "a="

COMMA ,

IDEN a

RPAR )

SEMI ;

RDKH }

INT int

IDEN add\_10

LPAR (

INT int

IDEN x

RPAR )

LDKH {

CONST const

INT int

IDEN ax

EQUAL =

UINT 10

SEMI ;

RETURN return

LPAR (

IDEN x

PLUS +

IDEN ax

RPAR )

SEMI ;

RDKH }

INT int

IDEN add\_a

LPAR (

INT int

IDEN x

RPAR )

LDKH {

INT int

IDEN aa

SEMI ;

IDEN aa

EQUAL =

UINT 112

SEMI ;

RETURN return

LPAR (

IDEN x

PLUS +

IDEN aa

RPAR )

SEMI ;

RDKH }

INT int

IDEN add\_1\_a

LPAR (

INT int

IDEN x

RPAR )

LDKH {

CONST const

INT int

IDEN ca

EQUAL =

UINT 13

SEMI ;

INT int

IDEN axx

SEMI ;

IDEN axx

EQUAL =

UINT 14

SEMI ;

RETURN return

LPAR (

IDEN x

PLUS +

IDEN ca

PLUS +

IDEN axx

RPAR )

SEMI ;

RDKH }

VOID void

IDEN nul

LPAR (

RPAR )

LDKH {

LDKH {

LDKH {

LDKH {

LDKH {

LDKH {

SEMI ;

RDKH }

RDKH }

RDKH }

RDKH }

RDKH }

SEMI ;

RDKH }

VOID void

MAIN main

LPAR (

RPAR )

LDKH {

CONST const

INT int

IDEN a0a

EQUAL =

UINT 1

SEMI ;

INT int

IDEN a1a

COMMA ,

IDEN a2a

COMMA ,

IDEN array11

LZKH [

UINT 1000

RZKH ]

COMMA ,

IDEN i

COMMA ,

IDEN j

COMMA ,

IDEN k

SEMI ;

CHAR char

IDEN cha

SEMI ;

IDEN a1a

EQUAL =

UINT 2

SEMI ;

IDEN a2a

EQUAL =

UINT 4

MINUS -

UINT 1

SEMI ;

IDEN i

EQUAL =

IDEN a1a

SEMI ;

IDEN array11

LZKH [

UINT 0

RZKH ]

EQUAL =

UINT 9

SEMI ;

IDEN n1

EQUAL =

MINUS -

UINT 2

SEMI ;

IDEN print\_all

LPAR (

IDEN n1

RPAR )

SEMI ;

IDEN N3

EQUAL =

PLUS +

MINUS -

UINT 4

SEMI ;

IDEN print\_all

LPAR (

IDEN N3

RPAR )

SEMI ;

IDEN N\_4

EQUAL =

MINUS -

PLUS +

UINT 5

SEMI ;

IDEN print\_all

LPAR (

IDEN N\_4

RPAR )

SEMI ;

IDEN array1

LZKH [

UINT 0

RZKH ]

EQUAL =

MINUS -

MINUS -

UINT 9

SEMI ;

IDEN print\_all

LPAR (

IDEN array

LZKH [

UINT 0

RZKH ]

RPAR )

SEMI ;

IDEN k

EQUAL =

MINUS -

IDEN array11

LZKH [

UINT 6

RZKH ]

PLUS +

IDEN n1

MULT \*

LPAR (

MINUS -

IDEN i

MINUS -

IDEN N3

DIV /

IDEN array1

LZKH [

UINT 0

RZKH ]

RPAR )

PLUS +

IDEN add

LPAR (

IDEN N\_4

COMMA ,

IDEN n\_2

RPAR )

SEMI ;

IDEN print\_all

LPAR (

IDEN k

RPAR )

SEMI ;

IDEN ch

EQUAL =

CHAR 'a'

SEMI ;

IDEN print\_char

LPAR (

IDEN ch

RPAR )

SEMI ;

IDEN str

LZKH [

UINT 1

RZKH ]

EQUAL =

CHAR 'A'

SEMI ;

IDEN print\_char

LPAR (

IDEN str

LZKH [

UINT 1

RZKH ]

RPAR )

SEMI ;

IDEN s\_0

LZKH [

UINT 2

RZKH ]

EQUAL =

CHAR '+'

SEMI ;

IDEN print\_char

LPAR (

IDEN s\_0

LZKH [

UINT 2

RZKH ]

RPAR )

SEMI ;

IDEN ch1

EQUAL =

CHAR '-'

SEMI ;

IDEN print\_char

LPAR (

IDEN ch1

RPAR )

SEMI ;

IDEN C2

EQUAL =

CHAR '\*'

SEMI ;

IDEN print\_char

LPAR (

IDEN C2

RPAR )

SEMI ;

IDEN c\_3

EQUAL =

CHAR '/'

SEMI ;

IDEN print\_char

LPAR (

IDEN c\_3

RPAR )

SEMI ;

IDEN Ch\_4

EQUAL =

CHAR '\_'

SEMI ;

IDEN print\_char

LPAR (

IDEN Ch\_4

RPAR )

SEMI ;

IF if

LPAR (

IDEN a0a

RPAR )

LDKH {

IDEN array11

LZKH [

UINT 2

RZKH ]

EQUAL =

IDEN array

LZKH [

UINT 0

RZKH ]

PLUS +

IDEN i

SEMI ;

IDEN print\_num

LPAR (

IDEN array11

LZKH [

UINT 2

RZKH ]

RPAR )

SEMI ;

RDKH }

IF if

LPAR (

IDEN a1a

ASSIGN ==

UINT 2

RPAR )

IDEN print\_num

LPAR (

IDEN a1a

PLUS +

UINT 1

RPAR )

SEMI ;

IDEN a1a

EQUAL =

IDEN a1a

PLUS +

UINT 1

SEMI ;

IF if

LPAR (

IDEN a1a

GREAT >

UINT 2

RPAR )

LDKH {

IDEN a1a

EQUAL =

IDEN a1a

MINUS -

IDEN a0a

SEMI ;

IDEN a1a

EQUAL =

IDEN a1a

MULT \*

UINT 2

SEMI ;

IDEN print\_num

LPAR (

IDEN a1a

RPAR )

SEMI ;

RDKH }

IF if

LPAR (

IDEN i

LESS <

UINT 10

RPAR )

LDKH {

IDEN i

EQUAL =

UINT 10

SEMI ;

IDEN print\_num

LPAR (

IDEN i

RPAR )

SEMI ;

RDKH }

IF if

LPAR (

IDEN i

GEQ >=

UINT 9

RPAR )

LDKH {

IDEN i

EQUAL =

UINT 1

SEMI ;

IDEN print\_num

LPAR (

IDEN i

RPAR )

SEMI ;

RDKH }

IF if

LPAR (

IDEN i

LEQ <=

UINT 9

RPAR )

LDKH {

IDEN i

EQUAL =

MINUS -

UINT 4

SEMI ;

IDEN print\_num

LPAR (

IDEN i

RPAR )

SEMI ;

RDKH }

IF if

LPAR (

IDEN i

NEQ !=

UINT 9

RPAR )

LDKH {

IDEN i

EQUAL =

PLUS +

UINT 9

SEMI ;

IDEN print\_num

LPAR (

IDEN i

RPAR )

SEMI ;

RDKH }

IDEN i

EQUAL =

IDEN i

MINUS -

UINT 9

SEMI ;

IF if

LPAR (

IDEN i

RPAR )

LDKH {

IDEN i

EQUAL =

UINT 9

SEMI ;

IDEN print\_num

LPAR (

IDEN i

RPAR )

SEMI ;

RDKH }

PRINTF printf

LPAR (

IDEN i

RPAR )

SEMI ;

IDEN j

EQUAL =

UINT 5

SEMI ;

IDEN while

LPAR (

IDEN i

LESS <

IDEN j

RPAR )

LDKH {

SWITCH switch

LPAR (

IDEN i

RPAR )

LDKH {

CASE case

UINT 0

COLON :

IDEN print\_num

LPAR (

IDEN i

RPAR )

SEMI ;

CASE case

UINT 1

COLON :

LDKH {

IDEN cha

EQUAL =

CHAR 'c'

SEMI ;

IDEN print\_char

LPAR (

IDEN cha

RPAR )

SEMI ;

RDKH }

CASE case

UINT 2

COLON :

IDEN print\_all

LPAR (

IDEN i

RPAR )

SEMI ;

CASE case

UINT 3

COLON :

IDEN print\_hello

SEMI ;

CASE case

UINT 4

COLON :

IDEN print\_all

LPAR (

IDEN i

RPAR )

SEMI ;

RDKH }

IDEN i

EQUAL =

IDEN i

PLUS +

UINT 1

SEMI ;

RDKH }

IDEN cha

EQUAL =

CHAR 'd'

SEMI ;

IDEN while

LPAR (

IDEN i

GREAT >

UINT 3

RPAR )

LDKH {

IDEN nul

LPAR (

RPAR )

SEMI ;

SWITCH switch

LPAR (

IDEN cha

RPAR )

LDKH {

CASE case

CHAR '2'

COLON :

IDEN print\_num

LPAR (

IDEN i

RPAR )

SEMI ;

CASE case

CHAR 'd'

COLON :

IDEN print\_all

LPAR (

IDEN i

RPAR )

SEMI ;

DEFAUL default

COLON :

IDEN print\_hello

LPAR (

RPAR )

SEMI ;

RDKH }

IDEN cha

EQUAL =

CHAR 'e'

SEMI ;

IDEN i

EQUAL =

IDEN i

MINUS -

UINT 1

SEMI ;

RDKH }

IDEN k

EQUAL =

IDEN add

LPAR (

IDEN i

PLUS +

IDEN j

RPAR )

SEMI ;

IDEN print\_all

LPAR (

IDEN k

RPAR )

SEMI ;

IDEN print\_all

LPAR (

IDEN add\_a

LPAR (

IDEN k

RPAR )

RPAR )

SEMI ;

IDEN print\_all

LPAR (

IDEN add\_10

LPAR (

IDEN k

RPAR )

RPAR )

SEMI ;

IDEN print\_all

LPAR (

IDEN add\_1\_a

LPAR (

IDEN k

RPAR )

RPAR )

SEMI ;

IDEN print\_char

LPAR (

IDEN getch

LPAR (

RPAR )

RPAR )

SEMI ;

IDEN print\_num

LPAR (

IDEN fac

LPAR (

UINT 5

RPAR )

RPAR )

SEMI ;

PRINTF printf

LPAR (

STRING "read i:"

RPAR )

SEMI ;

SCANF scanf

LPAR (

IDEN i

RPAR )

SEMI ;

IDEN print\_all

LPAR (

IDEN i

RPAR )

SEMI ;

PRINTF printf

LPAR (

STRING "read j,k:"

RPAR )

SEMI ;

SCANF scanf

LPAR (

IDEN j

COMMA ,

IDEN k

RPAR )

SEMI ;

IDEN print\_all

LPAR (

IDEN j

RPAR )

SEMI ;

IDEN print\_all

LPAR (

IDEN k

RPAR )

SEMI ;

IDEN print\_num

LPAR (

IDEN a

RPAR )

SEMI ;

IDEN print\_num

LPAR (

IDEN B

RPAR )

SEMI ;

IDEN print\_num

LPAR (

IDEN \_

RPAR )

SEMI ;

IDEN print\_num

LPAR (

IDEN \_d

RPAR )

SEMI ;

IDEN print\_char

LPAR (

IDEN c\_

RPAR )

SEMI ;

IDEN print\_char

LPAR (

IDEN c\_0

RPAR )

SEMI ;

IDEN print\_char

LPAR (

IDEN ch1

RPAR )

SEMI ;

IDEN print\_char

LPAR (

IDEN \_c\_2

RPAR )

SEMI ;

IDEN print\_char

LPAR (

IDEN C3

RPAR )

SEMI ;

IDEN print\_char

LPAR (

IDEN Ch\_4

RPAR )

SEMI ;

IDEN print\_char

LPAR (

IDEN \_C\_5\_

RPAR )

SEMI ;

RETURN return

SEMI ;

RDKH }

# 结论

根据测试程序“16061160\_test.txt”中的符合文法的源程序

生成名为“16061160\_result.txt”的词法分析结果

部分错误被输出到控制台。

结果表明词法分析正确。 **陈麒先**

2018/11/14