输入：文件的绝对路径

输出：

Input file name to compile:

CONSTSYM const

INTSYM int

IDENT length

BECOME =

NUM 10

COMMA ,

IDENT width

BECOME =

PLUS +

NUM 8

SEMICOLON ;

CONSTSYM const

CHARSYM char

IDENT firstLowerCase

BECOME =

ALPHA a

COMMA ,

IDENT firstUpperCase

BECOME =

ALPHA A

SEMICOLON ;

CONSTSYM const

CHARSYM char

IDENT lastLowerCase

BECOME =

ALPHA z

COMMA ,

IDENT lastUpperCase

BECOME =

ALPHA Z

SEMICOLON ;

INTSYM int

IDENT globalInt

SEMICOLON ;

INTSYM int

IDENT fib

LPARENTHESE (

INTSYM int

IDENT n

COMMA ,

INTSYM int

IDENT k

RPARENTHESE )

LBRACE {

INTSYM int

IDENT i

SEMICOLON ;

INTSYM int

IDENT res

SEMICOLON ;

IFSYM if

LPARENTHESE (

IDENT n

LEQ <=

IDENT k

RPARENTHESE )

LBRACE {

RETURNSYM return

LPARENTHESE (

NUM 1

RPARENTHESE )

SEMICOLON ;

RBRACE }

ELSESYM else

LBRACE {

IDENT res

BECOME =

NUM 0

SEMICOLON ;

FORSYM for

LPARENTHESE (

IDENT i

BECOME =

IDENT n

MINUS -

IDENT k

SEMICOLON ;

IDENT i

LESS <

IDENT n

SEMICOLON ;

IDENT i

BECOME =

IDENT i

PLUS +

NUM 1

RPARENTHESE )

LBRACE {

IDENT res

BECOME =

IDENT res

PLUS +

IDENT fib

LPARENTHESE (

IDENT i

COMMA ,

IDENT k

RPARENTHESE )

SEMICOLON ;

RBRACE }

RETURNSYM return

LPARENTHESE (

IDENT res

RPARENTHESE )

SEMICOLON ;

RBRACE }

RBRACE }

VOIDSYM void

IDENT print

LPARENTHESE (

RPARENTHESE )

LBRACE {

CHARSYM char

IDENT i

SEMICOLON ;

IDENT i

BECOME =

NUM 32

SEMICOLON ;

DOSYM do

LBRACE {

IFSYM if

LPARENTHESE (

IDENT i

NEQ !=

NUM 34

RPARENTHESE )

LBRACE {

PRINTFSYM printf

LPARENTHESE (

IDENT i

RPARENTHESE )

SEMICOLON ;

RBRACE }

IDENT i

BECOME =

IDENT i

PLUS +

NUM 1

SEMICOLON ;

RBRACE }

WHILESYM while

LPARENTHESE (

IDENT i

LESS <

NUM 127

RPARENTHESE )

RBRACE }

CHARSYM char

IDENT toUpperCase

LPARENTHESE (

CHARSYM char

IDENT x

RPARENTHESE )

LBRACE {

CHARSYM char

IDENT c

SEMICOLON ;

IFSYM if

LPARENTHESE (

IDENT x

GEQ >=

IDENT firstLowerCase

RPARENTHESE )

IFSYM if

LPARENTHESE (

IDENT x

LEQ <=

IDENT lastLowerCase

RPARENTHESE )

LBRACE {

IDENT c

BECOME =

IDENT x

MINUS -

IDENT firstLowerCase

PLUS +

IDENT firstUpperCase

SEMICOLON ;

RETURNSYM return

LPARENTHESE (

IDENT c

RPARENTHESE )

SEMICOLON ;

RBRACE }

ELSESYM else

RETURNSYM return

LPARENTHESE (

IDENT x

RPARENTHESE )

SEMICOLON ;

ELSESYM else

RETURNSYM return

LPARENTHESE (

IDENT x

RPARENTHESE )

SEMICOLON ;

RBRACE }

VOIDSYM void

IDENT changeGlobal

LPARENTHESE (

RPARENTHESE )

LBRACE {

PRINTFSYM printf

LPARENTHESE (

STR original:

RPARENTHESE )

SEMICOLON ;

PRINTFSYM printf

LPARENTHESE (

IDENT globalInt

RPARENTHESE )

SEMICOLON ;

IDENT globalInt

BECOME =

NUM 16

SEMICOLON ;

PRINTFSYM printf

LPARENTHESE (

STR changed:

RPARENTHESE )

SEMICOLON ;

PRINTFSYM printf

LPARENTHESE (

IDENT globalInt

RPARENTHESE )

SEMICOLON ;

RBRACE }

VOIDSYM void

IDENT dealRequest

LPARENTHESE (

INTSYM int

IDENT operation

RPARENTHESE )

LBRACE {

CONSTSYM const

CHARSYM char

IDENT x

BECOME =

ALPHA x

SEMICOLON ;

INTSYM int

IDENT i

SEMICOLON ;

CHARSYM char

IDENT y

SEMICOLON ;

PRINTFSYM printf

LPARENTHESE (

STR start

RPARENTHESE )

SEMICOLON ;

IDENT y

BECOME =

ALPHA y

SEMICOLON ;

IFSYM if

LPARENTHESE (

IDENT operation

EQU ==

NUM 0

RPARENTHESE )

LBRACE {

FORSYM for

LPARENTHESE (

IDENT i

BECOME =

NUM 1

SEMICOLON ;

IDENT i

LEQ <=

IDENT length

SEMICOLON ;

IDENT i

BECOME =

IDENT i

PLUS +

NUM 1

RPARENTHESE )

LBRACE {

PRINTFSYM printf

LPARENTHESE (

STR

COMMA ,

IDENT fib

LPARENTHESE (

IDENT i

COMMA ,

MINUS -

MINUS -

NUM 2

RPARENTHESE )

RPARENTHESE )

SEMICOLON ;

RBRACE }

RBRACE }

ELSESYM else

IFSYM if

LPARENTHESE (

IDENT operation

EQU ==

NUM 1

RPARENTHESE )

LBRACE {

IDENT print

LPARENTHESE (

RPARENTHESE )

SEMICOLON ;

RBRACE }

ELSESYM else

IFSYM if

LPARENTHESE (

IDENT operation

EQU ==

NUM 2

RPARENTHESE )

LBRACE {

PRINTFSYM printf

LPARENTHESE (

STR %c

COMMA ,

IDENT toUpperCase

LPARENTHESE (

IDENT x

RPARENTHESE )

RPARENTHESE )

SEMICOLON ;

PRINTFSYM printf

LPARENTHESE (

IDENT toUpperCase

LPARENTHESE (

ALPHA x

RPARENTHESE )

RPARENTHESE )

SEMICOLON ;

RBRACE }

ELSESYM else

IFSYM if

LPARENTHESE (

IDENT operation

EQU ==

NUM 3

RPARENTHESE )

LBRACE {

IDENT changeGlobal

LPARENTHESE (

RPARENTHESE )

SEMICOLON ;

RBRACE }

ELSESYM else

LBRACE {

PRINTFSYM printf

LPARENTHESE (

STR nothing

RPARENTHESE )

SEMICOLON ;

RBRACE }

RBRACE }

VOIDSYM void

MAINSYM main

LPARENTHESE (

RPARENTHESE )

LBRACE {

INTSYM int

IDENT i

SEMICOLON ;

INTSYM int

IDENT operation

SEMICOLON ;

IDENT globalInt

BECOME =

NUM 0

SEMICOLON ;

SCANFSYM scanf

LPARENTHESE (

IDENT operation

RPARENTHESE )

SEMICOLON ;

FORSYM for

LPARENTHESE (

IDENT i

BECOME =

NUM 0

SEMICOLON ;

IDENT i

LESS <

IDENT operation

SEMICOLON ;

IDENT i

BECOME =

IDENT i

PLUS +

NUM 1

RPARENTHESE )

LBRACE {

IDENT dealRequest

LPARENTHESE (

IDENT i

RPARENTHESE )

SEMICOLON ;

PRINTFSYM printf

LPARENTHESE (

STR

RPARENTHESE )

SEMICOLON ;

RBRACE }

RBRACE }

Lexicon analysing finished

分析：

与预期一致

注：

对于常量字符和字符串，输出去除了单双引号。