문법 분석 프로그램

주어진 문법에 따라 문법 분석 프로그램을 설계하고 구현하여 주십시오. 입출력 및 처리 요구사항은 다음과 같다:  
(1) 컴파일된 소스 파일은 자동 평가를 위해 testfile.txt, 출력된 결과 파일은 output.txt로 통일한다.결과 파일에는 다음 두 가지 정보가 포함되어 있습니다.  
    1) 단어의 식별 순서를 어법에 따라 분석하여 각 단어의 정보를 줄에 따라 출력한다(동어법 분석 작업을 요구하며 미리 읽었을 경우 출력할 수 없다).  
    2) 아래에 표시된 문법 분석 성분 분석이 끝나기 전에 현재 문법 성분의 이름을 "<상량 설명 >"( 비고: 출력을 요청하지 않은 문법 성분)과 같이 출력합니다.여전히 분석 필요)

이프로젝트는 제가 대학교 3학년 시절에 컴파일러 수업에서 나온 과제입니다.

정해진 문법을 토대로 c++언어를 사용하여 문법 분석 프로그램을 만드는겁니다.

먼저 词法分析（낱말분석)을 해야합니다.그다음 해야할게 语法分析（구문분석）입니다.

낱말분서은 제가 밑에 있는 이표를 보고 하였습니다

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 单词名称 | 类别码 | 单词名称 | 类别码 | 单词名称 | 类别码 | 单词名称 | 类别码 |
| 标识符 | IDENFR | if | IFTK | - | MINU | = | ASSIGN |
| 整型常量 | INTCON | else | ELSETK | \* | MULT | ; | SEMICN |
| 字符常量 | CHARCON | do | DOTK | / | DIV | , | COMMA |
| 字符串 | STRCON | while | WHILETK | < | LSS | ( | LPARENT |
| const | CONSTTK | for | FORTK | <= | LEQ | ) | RPARENT |
| int | INTTK | scanf | SCANFTK | > | GRE | [ | LBRACK |
| char | CHARTK | printf | PRINTFTK | >= | GEQ | ] | RBRACK |
| void | VOIDTK | return | RETURNTK | == | EQL | { | LBRACE |
| main | MAINTK | + | PLUS | != | NEQ | } | RBRACE |

낱말분석을 한뒤 밑에있는 문법을 보고 구문분석을 합니다.

최종적으로는 밑에있는 입력예시대로 입력하였을때에 결과값이 맞는지를 확인하고 스스로 다른 예시를 입력하여서 오류를 찾아내고 최대한 완벽하게 설계해야합니다.

＜加法运算符＞ ::= +｜-  
＜乘法运算符＞  ::= \*｜/  
＜关系运算符＞  ::=  <｜<=｜>｜>=｜!=｜==  
＜字母＞   ::= ＿｜a｜．．．｜z｜A｜．．．｜Z

＜数字＞   ::= ０｜＜非零数字＞  
＜非零数字＞  ::= １｜．．．｜９  
＜字符＞    ::=  '＜加法运算符＞'｜'＜乘法运算符＞'｜'＜字母＞'｜'＜数字＞'

＜字符串＞   ::=  "｛十进制编码为32,33,35-126的ASCII字符｝"

＜程序＞    ::= ［＜常量说明＞］［＜变量说明＞］{＜有返回值函数定义＞|＜无返回值函数定义＞}＜主函数＞

＜常量说明＞ ::=  const＜常量定义＞;{ const＜常量定义＞;}  
＜常量定义＞   ::=   int＜标识符＞＝＜整数＞{,＜标识符＞＝＜整数＞}

                  | char＜标识符＞＝＜字符＞{,＜标识符＞＝＜字符＞}

＜无符号整数＞  ::= ＜非零数字＞｛＜数字＞｝| 0  
＜整数＞        ::= ［＋｜－］＜无符号整数＞

＜标识符＞    ::=  ＜字母＞｛＜字母＞｜＜数字＞｝

＜声明头部＞   ::=  int＜标识符＞ |char＜标识符＞

＜变量说明＞  ::= ＜变量定义＞;{＜变量定义＞;}

＜变量定义＞  ::= ＜类型标识符＞(＜标识符＞|＜标识符＞'['＜无符号整数＞']'){,(＜标识符＞|＜标识符＞'['＜无符号整数＞']' )}

                 //＜无符号整数＞表示数组元素的个数，其值需大于0

＜类型标识符＞      ::=  int | char

＜有返回值函数定义＞  ::=  ＜声明头部＞'('＜参数表＞')' '{'＜复合语句＞'}'  
＜无返回值函数定义＞  ::= void＜标识符＞'('＜参数表＞')''{'＜复合语句＞'}'  
＜复合语句＞   ::=  ［＜常量说明＞］［＜变量说明＞］＜语句列＞

＜参数表＞    ::=  ＜类型标识符＞＜标识符＞{,＜类型标识符＞＜标识符＞}| ＜空＞  
＜主函数＞    ::= void main‘(’‘)’ ‘{’＜复合语句＞‘}’

＜表达式＞    ::= ［＋｜－］＜项＞{＜加法运算符＞＜项＞}   //[+|-]只作用于第一个<项>  
＜项＞     ::= ＜因子＞{＜乘法运算符＞＜因子＞}  
＜因子＞    ::= ＜标识符＞｜＜标识符＞'['＜表达式＞']'|'('＜表达式＞')'｜＜整数＞|＜字符＞｜＜有返回值函数调用语句＞           
＜语句＞    ::= ＜条件语句＞｜＜循环语句＞| '{'＜语句列＞'}'| ＜有返回值函数调用语句＞;   
                           |＜无返回值函数调用语句＞;｜＜赋值语句＞;｜＜读语句＞;｜＜写语句＞;｜＜空＞;|＜返回语句＞;

＜赋值语句＞   ::=  ＜标识符＞＝＜表达式＞|＜标识符＞'['＜表达式＞']'=＜表达式＞  
＜条件语句＞  ::= if '('＜条件＞')'＜语句＞［else＜语句＞］  
＜条件＞    ::=  ＜表达式＞＜关系运算符＞＜表达式＞ //整型表达式之间才能进行关系运算

       ｜＜表达式＞    //表达式为整型，其值为0条件为假，值不为0时条件为真

＜循环语句＞   ::=  while '('＜条件＞')'＜语句＞| do＜语句＞while '('＜条件＞')' |for'('＜标识符＞＝＜表达式＞;＜条件＞;＜标识符＞＝＜标识符＞(+|-)＜步长＞')'＜语句＞  
＜步长＞::= ＜无符号整数＞    
＜有返回值函数调用语句＞ ::= ＜标识符＞'('＜值参数表＞')'  
＜无返回值函数调用语句＞ ::= ＜标识符＞'('＜值参数表＞')'  
＜值参数表＞   ::= ＜表达式＞{,＜表达式＞}｜＜空＞  
＜语句列＞   ::= ｛＜语句＞｝  
＜读语句＞    ::=  scanf '('＜标识符＞{,＜标识符＞}')'  
＜写语句＞    ::= printf '(' ＜字符串＞,＜表达式＞ ')'| printf '('＜字符串＞ ')'| printf '('＜表达式＞')'  
＜返回语句＞   ::=  return['('＜表达式＞')']

【样例输入】//입력예시

const int const1 = 1, const2 = -100;

const char const3 = '\_';

int change1;

char change3;

int gets1(int var1,int var2){

    change1 = var1 + var2;

    return (change1);

}

void main(){

    printf("Hello World");

    printf(gets1(10, 20));

}

【样例输出】//출력예시

CONSTTK const

INTTK int

IDENFR const1

ASSIGN =

INTCON 1

<无符号整数>

<整数>

COMMA ,

IDENFR const2

ASSIGN =

MINU -

INTCON 100

<无符号整数>

<整数>

<常量定义>

SEMICN ;

CONSTTK const

CHARTK char

IDENFR const3

ASSIGN =

CHARCON \_

<常量定义>

SEMICN ;

<常量说明>

INTTK int

IDENFR change1

<变量定义>

SEMICN ;

CHARTK char

IDENFR change3

<变量定义>

SEMICN ;

<变量说明>

INTTK int

IDENFR gets1

<声明头部>

LPARENT (

INTTK int

IDENFR var1

COMMA ,

INTTK int

IDENFR var2

<参数表>

RPARENT )

LBRACE {

IDENFR change1

ASSIGN =

IDENFR var1

<因子>

<项>

PLUS +

IDENFR var2

<因子>

<项>

<表达式>

<赋值语句>

SEMICN ;

<语句>

RETURNTK return

LPARENT (

IDENFR change1

<因子>

<项>

<表达式>

RPARENT )

<返回语句>

SEMICN ;

<语句>

<语句列>

<复合语句>

RBRACE }

<有返回值函数定义>

VOIDTK void

MAINTK main

LPARENT (

RPARENT )

LBRACE {

PRINTFTK printf

LPARENT (

STRCON Hello World

<字符串>

RPARENT )

<写语句>

SEMICN ;

<语句>

PRINTFTK printf

LPARENT (

IDENFR gets1

LPARENT (

INTCON 10

<无符号整数>

<整数>

<因子>

<项>

<表达式>

COMMA ,

INTCON 20

<无符号整数>

<整数>

<因子>

<项>

<表达式>

<值参数表>

RPARENT )

<有返回值函数调用语句>

<因子>

<项>

<表达式>

RPARENT )

<写语句>

SEMICN ;

<语句>

<语句列>

<复合语句>

RBRACE }

<主函数>

<程序>

코드: