<程序> ::= <分程序>.

<分程序> ::= [<常量说明部分>][<变量说明部分>]{[<过程说明部分>]| [<函数说明部分>]}<复合语句>

<常量说明部分> ::= const<常量定义>{,<常量定义>};

<常量定义> ::= <标识符>=<常量>

<常量> ::= [+| -]<无符号整数>|<字符>

<字符> ::= '<字母>' | '<数字>'

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

<无符号整数> ::= <数字>{<数字>}

<标识符> ::= <字母>{<字母>|<数字>}

<变量说明部分> ::= var <变量说明> ; {<变量说明>;}

<变量说明> ::= <标识符>{, <标识符>} : <类型>

<类型> ::= <基本类型>|array'['<无符号整数>']' of <基本类型>

<基本类型> ::= integer | char

<过程说明部分> ::= <过程首部><分程序>{;<过程首部><分程序>};

<函数说明部分> ::= <函数首部><分程序>{;<函数首部><分程序>};

<过程首部> ::= procedure<标识符>[<形式参数表>];

<函数首部> ::= function <标识符>[<形式参数表>]:

<基本类型>;

<形式参数表> ::= '(' <形式参数段>{; <形式参数段>}')'

<形式参数段> ::= [var]<标识符>{, <标识符>}: <基本类型>

<语句> ::= <赋值语句>|<条件语句>|<情况语句>|<过程调用语句>|<复合语句>|<读语句>|<写语句>|<for 循环语句>|<空>

<赋值语句> ::= <标识符> := <表达式>| <函数标识符> := <表达式> | <标识符>'['<表达式>']':= <表达式>

<函数标识符> ::= <标识符>

<表达式> ::= [+|-]<项>{<加法运算符><项>}

<项> ::= <因子>{<乘法运算符><因子>}

<因子> ::= <标识符>|<无符号整数>|'('<表达式>')' |<函数调用语句>|<标识符>'['<表达式>']'

<函数调用语句> ::= <标识符>[<实在参数表>]

<实在参数表> ::= '('<实在参数> {, <实在参数>}')'

<实在参数> ::= <表达式>

<加法运算符> ::= +|-

<乘法运算符> ::= *|/

<条件> ::= <表达式><关系运算符><表达式>

<关系运算符> ::= <|<=|>|>= |=|<>

<条件语句> ::= if<条件>then<语句> | if<条件>then<语句>else<语句>

<情况语句> ::= case <表达式> of <情况表元素>{; <情况表元素>}end

<情况表元素> ::= <常量> : <语句>

<for 循环语句> ::= for <标识符> := <表达式>（downto | to） <表达式> do <语句> //步长为 1

<过程调用语句> ::= <标识符>[<实在参数表>]

<复合语句> ::= begin<语句>{; <语句>}end

<读语句> ::= read'('<标识符>{,<标识符>}')'

<写语句> ::= write '(' <字符串>,<表达式>')'|write'(' <字符串> ')'|write'('<表达式>')'

<字母> ::= a|b|c|d…x|y|z |A|B…|Z

<数字> ::= 0|1|2|3…8|9

附加说明：

（1）char 类型的变量或常量，用字符的 ASCII 码对应的整数参加运算

（2）标识符不区分大小写字母

（3）赋值语句中<函数标识符> := <表达式> 作为函数的返回值，其类型应与返回类型一致，此语句后面的语句可继续执行

（4）写语句中的字符串原样输出，表达式只有单个字符类型的

变量或常量按字符输出，其他表达式均按整型输出

（5）情况语句中，case 后面的表达式和情况表元素里面的常量只允许出现 integer 和 char 类型

（6）数组的下标从 0 开始

（7）带 var 的形式参数为变量形参，实参与该类形参传递数据时是传地址