测试结果分析

测试1（正确）：

在此测试重点是在测试程序对生成目标代码，结果是正确，编译程序能正确生成目标代码。

测试2（错误）：

在此测试重点是在测试程序对RED WRT目标代码生成，结果是错误，因为程序源代码中本恶人忘了加 RED 和WRT 的P-code。

测试3（正确）：

在此测试重点是在测试程序对分析P-code的指令，结果是正确，编译程序能正确生成P-code指令。

测试4（正确）：

在此测试重点是在测试程序对源代码的错误处理，结果是正确，但编译程序一到出现fatal error 它直接退出，而程序会告诉我们错误的号码。

测试5（正确）：

在此测试重点是在测试程序对每个procedure的代码，程序能输出每个procedure的代码，但一扫描到不合法的文法，程序退出。

测试6（错误）

在此测试重点是在控制然后有一点小问题是在在控制台输入的带路径的文件名，打开文件名对应的文件进行编译。文件会找不到或者都不到，可能原因是在输入路径的时候不对，或者本人用的函数不太合适。结果错误。

这程序在判断RED 和WRT 的还有问题。然后有一点小问题是在在控制台输入的带路径的文件名，打开文件名对应的文件进行编译。

本人获取的文法是如此：

<程序>                    ::=  <分程序>.  
<分程序>               ::=  [<变量说明部分>][<常量说明部分>][<过程说明部分>]<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>=<无符号整数>  
<无符号整数>       ::=  <非零数字>{<数字>}|0  
<标识符>                ::=  <字母>{<字母>|<数字>}  
<变量说明部分>   ::=  var<标识符>{, <标识符>};  
<过程说明部分>   ::=  <过程首部><分程序>{;<过程首部><分程序>};  
<过程首部>            ::=  procedure<标识符>;  
<语句>                    ::=  <赋值语句>|<条件语句>|<当循环语句>|<过程调用语句>|<复合语句>|<读语句>|<写语句>|<空>   
<赋值语句>           ::=  <标识符> := <表达式>  
<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>|<无符号整数>|'(' <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>|odd<表达式>  
<关系运算符>       ::=  =|<>|<|<=|>|>=  
<条件语句>           ::=  if<条件>then<语句>  
<当循环语句>       ::=  while<条件>do<语句>   
<过程调用语句>  ::=  call<标识符>   
<复合语句>           ::=  begin<语句>{;<语句>}end  
<读语句>               ::=  read'('<标识符>{, <标识符>}')'   
<写语句>               ::=  write'('<表达式>{, <表达式>}')'  
<字母>                    ::=  a|b|c|d...|x|y|z|A|B...|Z   
<数字>                    ::=  0|<非零数字>  
<非零数字>           ::=  1|2|3...|8|9

下面是程序的测试分析结果：





