**项目测试报告**

**任务一**

**1、准备工作**

**tiny的正则表达式：**

|  |
| --- |
| if|then|else|end|repeat|until|read|write  \+\ | - | \\*\ | / | % | < | <> | <= | >= | > | = | ; | := | \(\ | \)\  PLUS | MINUS | MULTIPLY | DIVIDE | MOD | LT| NE | LTEQ | RTEQ | RT | EQ | SEMI | ASSIGN | LPAN | RPAN  letter(letter|num)\*  num num\*  {~} |

**sample.tny文件：**

|  |
| --- |
| read xXx; {read和标识符判断}  if ( xXx < yYy ) then {if和<判断}  repeat{repeat判断}  xXx := (xXx + 1) \* 1 - 0 /1{+-\*/的判断}  until xXx = yYy{until和=判断}  else{else判断}  write 001+123 % 2{%和write、数字的判断}  end;{end判断}  read z1;  {各种符号判断}  if z1 <= 1000 then  z1 := 0  else  z1 := 1  end;  if z1 <> 1000 then  z1 := 0  else  z1 := 1  end;  if z1 >= 1000 then  z1 := 0  else  z1 := 1  end;  if z1 > 1000 then  z1 := 0  else  z1 := 1  end |

**输入正则表达式：**



**2、NFA**



**3、DFA图**



**4、DFA图最小化**



**5、生成词法程序**

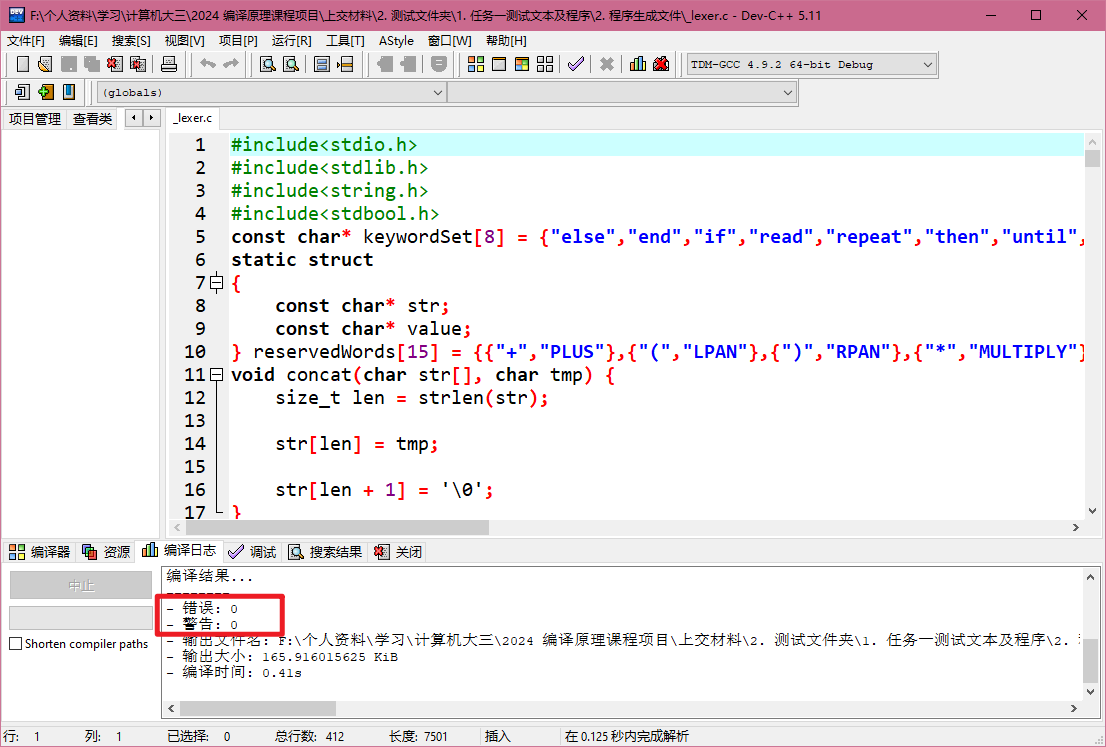


代码太长，具体可查看：

2. 测试文件夹\1. 任务一测试文本及程序\2. 程序生成文件中的\_lexer.c文件

**6、编译\_lexer.c文件并运行**

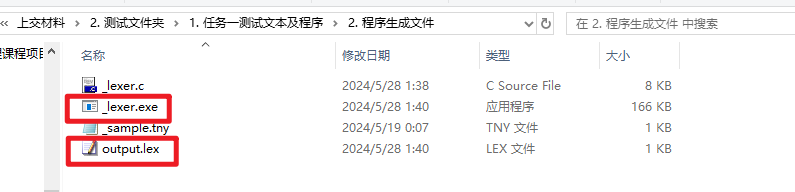
**注意：sample.tny必须和该程序放在同一个文件夹下。**



上图可知：编译成功

并成功生成lex文件：

**7、查看lex文件**





具体lex如下：

|  |
| --- |
| read:read  ID:xXx  SEMI:;  if:if  LPAN:(  ID:xXx  LT:<  ID:yYy  RPAN:)  then:then  repeat:repeat  ID:xXx  ASSIGN::=  LPAN:(  ID:xXx  PLUS:+  NUMBER:1  RPAN:)  MULTIPLY:\*  NUMBER:1  MINUS:-  NUMBER:0  DIVIDE:/  NUMBER:1  until:until  ID:xXx  EQ:=  ID:yYy  else:else  write:write  NUMBER:001  PLUS:+  NUMBER:123  MOD:%  NUMBER:2  end:end  SEMI:;  read:read  ID:z1  SEMI:;  if:if  ID:z1  LTEQ:<=  NUMBER:1000  then:then  ID:z1  ASSIGN::=  NUMBER:0  else:else  ID:z1  ASSIGN::=  NUMBER:1  end:end  SEMI:;  if:if  ID:z1  NE:<>  NUMBER:1000  then:then  ID:z1  ASSIGN::=  NUMBER:0  else:else  ID:z1  ASSIGN::=  NUMBER:1  end:end  SEMI:;  if:if  ID:z1  RTEQ:>=  NUMBER:1000  then:then  ID:z1  ASSIGN::=  NUMBER:0  else:else  ID:z1  ASSIGN::=  NUMBER:1  end:end  SEMI:;  if:if  ID:z1  RT:>  NUMBER:1000  then:then  ID:z1  ASSIGN::=  NUMBER:0  else:else  ID:z1  ASSIGN::=  NUMBER:1  end:end  EOF:EOF |

对照tiny源程序，可知解析完全正确。

**测试结果**

任务一测试完全通过