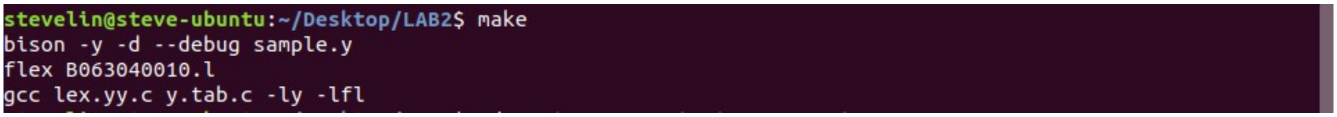
ReadMe

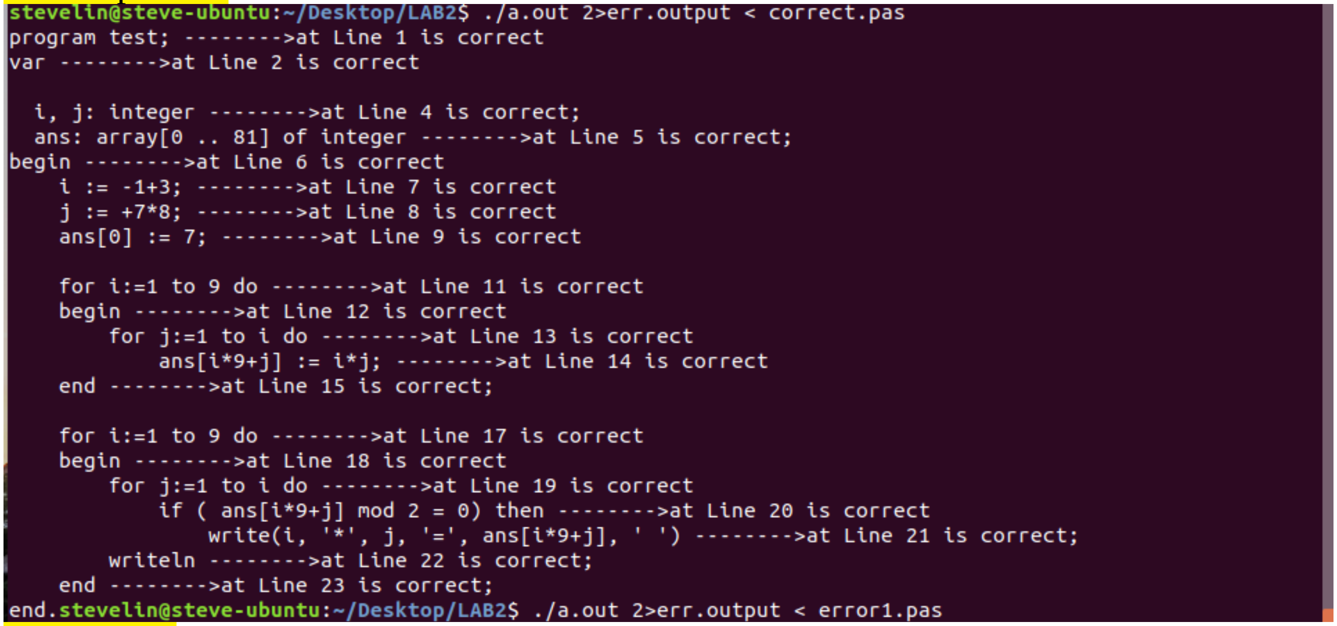
makefile



```
steveLin@steve-ubuntu:~/Desktop/LAB2$ make
bison -y -d --debug sample.y
flex B063040010.l
gcc lex.yy.c y.tab.c -ly -lfl
```

lanch ./a.out

correct.pascal



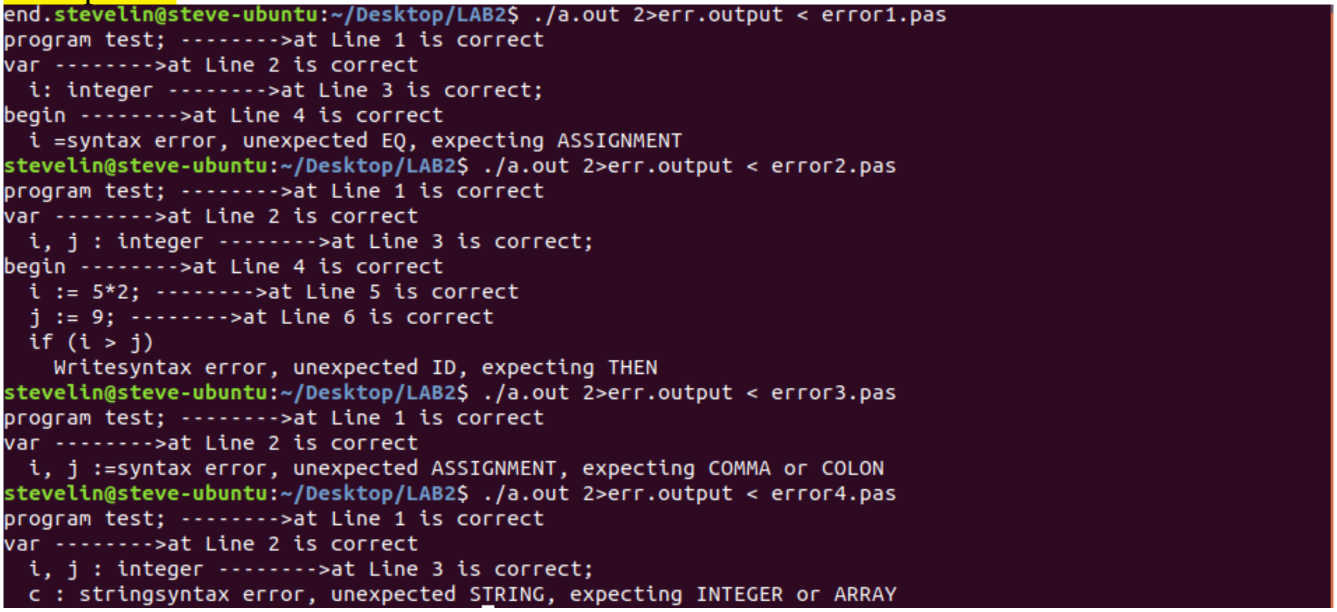
```
steveLin@steve-ubuntu:~/Desktop/LAB2$ ./a.out 2>err.output < correct.pas
program test; -------->at Line 1 is correct
var -------->at Line 2 is correct

  i, j: integer -------->at Line 4 is correct;
  ans: array[0 .. 81] of integer -------->at Line 5 is correct;
begin -------->at Line 6 is correct
    i := -1+3; -------->at Line 7 is correct
    j := +7*8; -------->at Line 8 is correct
    ans[0] := 7; -------->at Line 9 is correct

    for i:=1 to 9 do -------->at Line 11 is correct
    begin -------->at Line 12 is correct
        for j:=1 to i do -------->at Line 13 is correct
            ans[i*9+j] := i*j; -------->at Line 14 is correct
    end -------->at Line 15 is correct;

    for i:=1 to 9 do -------->at Line 17 is correct
    begin -------->at Line 18 is correct
        for j:=1 to i do -------->at Line 19 is correct
            if ( ans[i*9+j] mod 2 = 0) then -------->at Line 20 is correct
                write(i, '*', j, '=', ans[i*9+j], ' ') -------->at Line 21 is correct;
        writeln -------->at Line 22 is correct;
    end -------->at Line 23 is correct;
end.steveLin@steve-ubuntu:~/Desktop/LAB2$ ./a.out 2>err.output < error1.pas
```

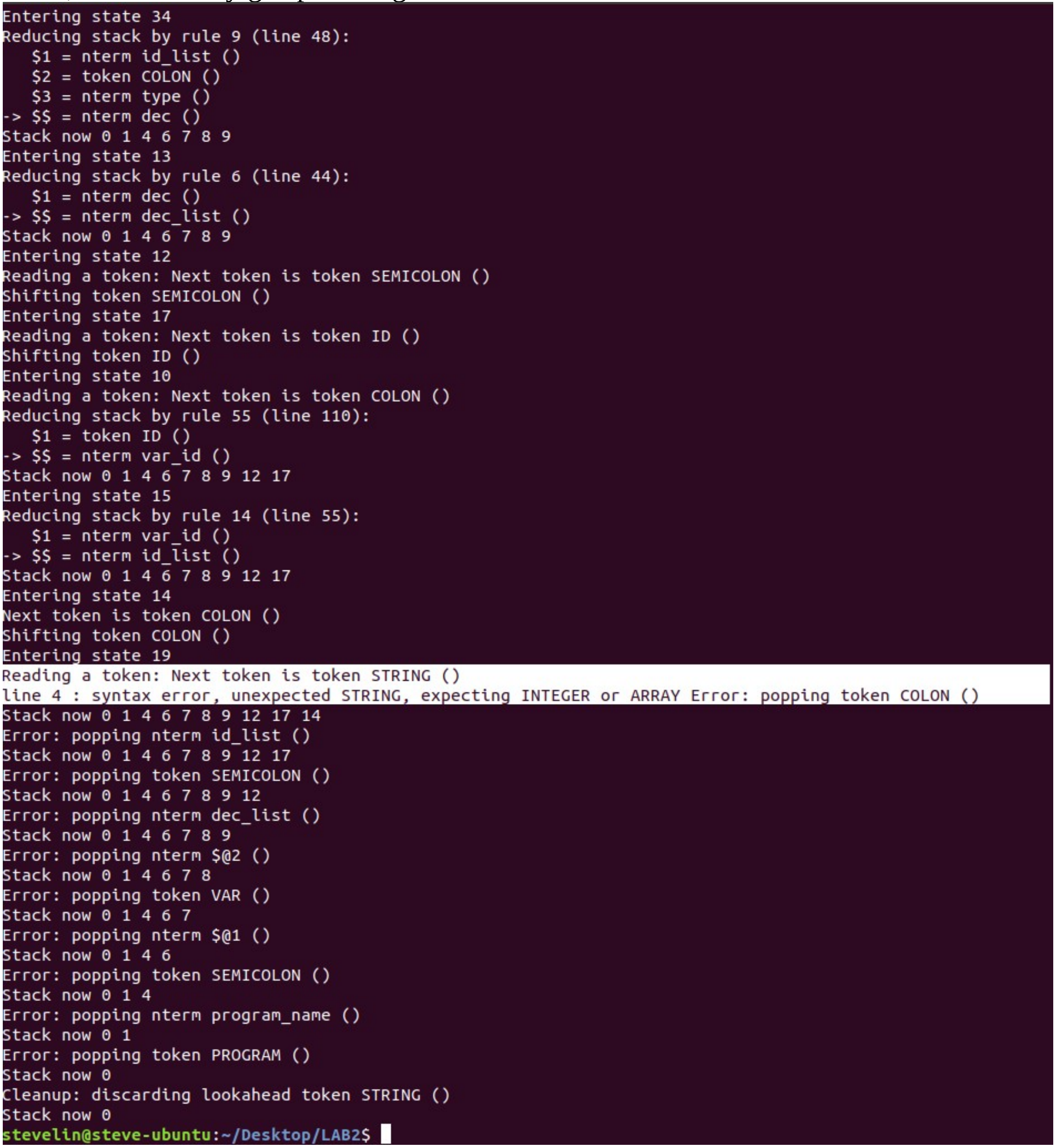
error.pascal



```
end.steveLin@steve-ubuntu:~/Desktop/LAB2$ ./a.out 2>err.output < error1.pas
program test; -------->at Line 1 is correct
var -------->at Line 2 is correct
  i: integer -------->at Line 3 is correct;
begin -------->at Line 4 is correct
  i =syntax error, unexpected EQ, expecting ASSIGNMENT
steveLin@steve-ubuntu:~/Desktop/LAB2$ ./a.out 2>err.output < error2.pas
program test; -------->at Line 1 is correct
var -------->at Line 2 is correct
  i, j : integer -------->at Line 3 is correct;
begin -------->at Line 4 is correct
  i := 5*2; -------->at Line 5 is correct
  j := 9; -------->at Line 6 is correct
  if (i > j)
    Writesyntax error, unexpected ID, expecting THEN
steveLin@steve-ubuntu:~/Desktop/LAB2$ ./a.out 2>err.output < error3.pas
program test; -------->at Line 1 is correct
var -------->at Line 2 is correct
  i, j :=syntax error, unexpected ASSIGNMENT, expecting COMMA or COLON
steveLin@steve-ubuntu:~/Desktop/LAB2$ ./a.out 2>err.output < error4.pas
program test; -------->at Line 1 is correct
var -------->at Line 2 is correct
  i, j : integer -------->at Line 3 is correct;
  c : stringsyntax error, unexpected STRING, expecting INTEGER or ARRAY
```

why use std_errror ???

it’s not only std_error, bison offer a excellent option function that give you detail of the process how bison compile yacc file. By reading the output of std_error generate by bison, we can easily grasp the bug at alance.



```
Entering state 34
Reducing stack by rule 9 (line 48):
   $1 = nterm id_list ()
   $2 = token COLON ()
   $3 = nterm type ()
-> $$ = nterm dec ()
Stack now 0 1 4 6 7 8 9
Entering state 13
Reducing stack by rule 6 (line 44):
   $1 = nterm dec ()
-> $$ = nterm dec_list ()
Stack now 0 1 4 6 7 8 9
Entering state 12
Reading a token: Next token is token SEMICOLON ()
Shifting token SEMICOLON ()
Entering state 17
Reading a token: Next token is token ID ()
Shifting token ID ()
Entering state 10
Reading a token: Next token is token COLON ()
Reducing stack by rule 55 (line 110):
   $1 = token ID ()
-> $$ = nterm var_id ()
Stack now 0 1 4 6 7 8 9 12 17
Entering state 15
Reducing stack by rule 14 (line 55):
   $1 = nterm var_id ()
-> $$ = nterm id_list ()
Stack now 0 1 4 6 7 8 9 12 17
Entering state 14
Next token is token COLON ()
Shifting token COLON ()
Entering state 19
Reading a token: Next token is token STRING ()
line 4 : syntax error, unexpected STRING, expecting INTEGER or ARRAY Error: popping token COLON ()
Stack now 0 1 4 6 7 8 9 12 17 14
Error: popping nterm id_list ()
Stack now 0 1 4 6 7 8 9 12 17
Error: popping token SEMICOLON ()
Stack now 0 1 4 6 7 8 9 12
Error: popping nterm dec_list ()
Stack now 0 1 4 6 7 8 9
Error: popping nterm $@2 ()
Stack now 0 1 4 6 7 8
Error: popping token VAR ()
Stack now 0 1 4 6 7
Error: popping nterm $@1 ()
Stack now 0 1 4 6
Error: popping token SEMICOLON ()
Stack now 0 1 4
Error: popping nterm program_name ()
Stack now 0 1
Error: popping token PROGRAM ()
Stack now 0
Cleanup: discarding lookahead token STRING ()
Stack now 0
stevelin@steve-ubuntu:~/Desktop/LAB2$
```