```
/*write a yak specification to check the syntax of while loop. Also count the
level of nesitng*/

//LEX program
%{
        #include "y.tab.h"
%}

%%

"while" {return WHILE;}
[a-z]+  {return id;}
"+" {return '+';}
"-" {return '-';}
"*" {return '*';}
"/" {return '/';}
"(" {return '(';}
")" {return ')';}
">" {return GT;}
"<" {return LT;}
">=" {return GTE;}
"==" {return EQ;}
"<=" {return LTE;}
"&&" {return AND;}
"||" {return OR;}
";" {return ';';}
"{" {return '{';}
"}" {return '}';}
[ \t] ;
[0-9]+ {yylval=atoi(yytext);return num;}
%%

int yywrap() {
    return 1;
}


//YACC program
%{
   #include<stdio.h>
   #include<stdlib.h>
   void yyerror();
   int cnt=0 ;
%}

%token num id '+' '-' '*' '/' WHILE GT LT EQ LTE GTE AND OR
%left GT LT EQ LTE GTE
%left AND OR
%left '+' '-'
%left '*' '/'
%left '(' ')'

%%


S   : ST {printf("valid count= %d",cnt);} ;
ST  : WHILE '(' E2 ')' '{' DEF '}' {cnt++;} ;
DEF :  E
    | ST
    ;
E   : id '=' E
    | E '+' E
    | E '-' E
    | E '*' E
    | E '/' E
    | id
    | num
    | ';' ;
BODY: E   ;
E2  : E AND E
    | E OR E
    | E LT E
    | E GT E
    | E GTE E
    | E LTE E
    | E EQ E
    | E2 AND E2
    | E2 OR E2
    | E

     ;
%%


int main() {
   printf("write a while loop:\t");
   yyparse();
}

void
yyerror (char const *s)
{
  fprintf (stderr, "%s\n", s);
}

/*OUTPUT
paddi@jarvis:~/Desktop/LP/exp2$ lex prac3.l
paddi@jarvis:~/Desktop/LP/exp2$ yacc -d prac3.y
paddi@jarvis:~/Desktop/LP/exp2$ gcc lex.yy.c y.tab.c -o a.out
paddi@jarvis:~/Desktop/LP/exp2$ ./a.out
write a while loop:     while(a<b && c<b){while(b<d){a}}
valid count= 2

*/
```