1. To Validate nested for, nested while using lex and yacc.

Lex Code

```
%{
        #include<stdio.h>
        #include "lab10s.tab.h"
        void yyerror(char*);
%}

%%
"for" {
yylval=yytext[0];
return VARIABLE;
}
"while" {
yylval=yytext[0];
return VARIABLE;
}
[{] {
yylval=yytext[0];
return START;
}
[}] {
yylval=yytext[0];
return END;
}
[\n] return *yytext;
. ;
%%
int yywrap(void)
{
return 1;
}
```

Yacc Code

```
%{
        #include<stdio.h>
        int yylex(void);
        void yyerror(char*);
%}
%token INTEGER VARIABLE SPECIAL START END
%%
program:
        program statement '\n' { printf("VALID SYNTAX\n");}
        |
        ;
statement:
        letter START END{}
          | letter START statement END {}
        ;
letter:
        VARIABLE

        ;

%%
void yyerror(char*s)
{
printf("%s",s);
}
int main()
{
yyparse();
return 0;
}
```

Output

```
C:\Flex Windows\EditPlusPortable>lex lab10s.l

C:\Flex Windows\EditPlusPortable>yacc -d lab10s.y

C:\Flex Windows\EditPlusPortable>cc lex.yy.c lab10s.tab.c

C:\Flex Windows\EditPlusPortable>a.exe
for(int i=0;i<5;i++){ for(int j=0;j<i;j++){printf("%d",i*j);}}
VALID SYNTAX
int i=0; while(i<5){ int j=0; while(j<i){ printf("%d",j); j++;}printf("%d",i*j) i++;}
VALID SYNTAX


C:\Flex Windows\EditPlusPortable>
```