Lab 5: Write a program to check syntax of declaration statement using LEX and YACC.

declaration.y

```
%{
#include <stdio.h>
%}
%token Int Char Float Bool String IntV CharV FloatV BoolV StringV Id Am Const
%%
S1: S1 S
 | S
 ;
S: Int Iv';' { printf("int declaration accepted"); }
| Char Cc';' { printf("char declaration accepted"); }
| Float Ff';' { printf("float declaration accepted"); }
| Bool Bb';' { printf("bool declaration accepted."); }
;
Iv: IdM
| Iv ',' Id
| Id '=' IntV
| Iv ',' Id '=' IntV
;
Cc: IdM
| Cc ',' Id
| Id '=' CharV
| Cc ',' Id '=' CharV
;
Ff: IdM
| Ff ',' Id
| Id '=' FloatV
| Ff ',' Id '=' FloatV
;
Bb: IdM
```

```
| Bb ',' Id
| Id '=' BoolV
| Bb ',' Id '=' BoolV
;
Ss: IdM
| Ss ',' Id
| Id '=' StringV
| Ss ',' Id '=' StringV
;
IdM: Id
;
%%
void yyerror(char*s) {
printf("%s", s);
}
int main() {
yyparse();
return 0;
}
```

Declaration.l

```
%{
#include "y.tab.h"
%}
letter [a-zA-Z]
num [0-9]+
float {num}+\.{num}+
bools "true"|"false"
identifier {letter}({letter}|{num})*
%%
"int" { return Int; }
```

"char" { return Char; }

"float" { return Float; }

"bool" { return Bool; }

{num} { return IntV; }

{float} { return FloatV; }

{bools} { return BoolV; }

{identifier} { return Id; }

"'"(.)"'" { return CharV; }

[,;=] { return yytext[0]; }

%%

int yywrap() {

return 1;

}

OUTPUT: