Exercise 6 - Implementation of Syntax checker using Lex Yacc Tools

Harshini S
185001058

Program Code:

syntax.l

```
%{
#include<stdio.h>
#include<stdlib.h>
#include "y.tab.h"
void yyerror(char *);
extern int yylval;
int lineNo = 1;
%}

stringid ([_a-zA-Z][_a-zA-Z0-9]*\[[0-9]*\])
num ([0-9\.]+)
string (\".*\")
character (\'.*\')

%%
("++"|"--") {return UNARY;}
("<"|"<="|">"|">="|"=="|"!=") { return RELOP; }
("+"|"-"|"*"|"/"|"%"|"<<"|">>"|"!"|"&&"|"||") { return
OP; } ("+="|"-="|"*="|"/="|"=") {return ASSIG_OP;}
("int"|"float"|"char"|"double") { return DTYPE; }
if {return IF;}
else {return ELSE;}
while {return WHILE;}
for {return FOR;}

[_a-zA-Z][_a-zA-Z0-9]* { return ID; }
{stringid} { return STRING_ID; }
{num} { return NUM; }
{string} {return STRING;}
{character} { return CHAR;}
[ \t]+ { }
[ \n] {lineNo+=1;}
(";"|"\."|"("|")"|"{"|"}") {return *yytext;}
. {
  fprintf(stderr,"Unknown token found: <%s>\n",
yytext); }
%%
```

syntax.y

```
%{
  #include<stdio.h>
  #include<stdlib.h>
  #include<math.h>
  #include"y.tab.h"
  extern int lineNo;
  int yylex(void);
  void yyerror();
  int flag=0;

%}

%token NUM STRING CHAR ID STRING_ID OP ASSIG_OP DTYPE RELOP IF
ELSE WHILE FOR UNARY

%%
program: statement program
       | statement
statement: declaration';'
          | assignment';'
          | expr';'
          | conditional
          | loop
declaration: DTYPE ID
              | DTYPE STRING_ID
              | DTYPE ID ASSIG_OP expr
              | DTYPE STRING_ID ASSIG_OP expr
assignment: ID ASSIG_OP expr
conditional: IF expr statement ELSE statement
            | IF expr statement ELSE '{' program '}'
            | IF expr '{' program '}' ELSE statement
            | IF expr '{' program '}' ELSE '{' program '}'
            | IF expr statement
            | IF expr '{' program '}'

loop: WHILE '(' expr ')' '{' program '}'
    | WHILE '(' assignment ')' '{' program '}'
    | FOR '(' DTYPE ID ASSIG_OP expr ';' expr ';' expr ')'
| FOR '(' DTYPE ID ASSIG_OP expr ';' expr ';' expr ')' '{'
program '}'
    | FOR '(' ID ASSIG_OP expr ';' expr ';' expr ')' statement |
    FOR '(' ID ASSIG_OP expr ';' expr ';' expr ')' '{' program
    '}'

expr: expr OP expr
    | expr RELOP expr
    | '(' expr ')'
    | ID UNARY
    | ID | NUM | STRING | CHAR
%%
void yyerror(){
    fprintf(stderr, "Invalid syntax at Line No : ");
    printf("%d\n",lineNo);
    flag = 1;
    return;
}
int yywrap(){
    return 1;
}
int main()
{
    yyparse();
    if(!flag){
        printf("Valid syntax\n");
    }
    return 0;
}
```

input.txt

```
int i, a;
i = 0;
a=5;
while (i < 10)
{
 if ( i<a)
i = i + 1;
 else
 i = i – 1;
}
```