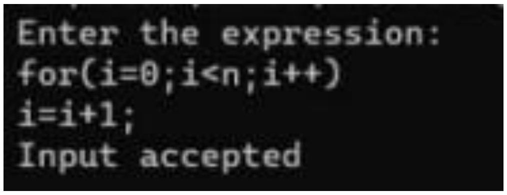
Antriksh Sharma
20070122021
CCL - LEX & YACC

Input & Output

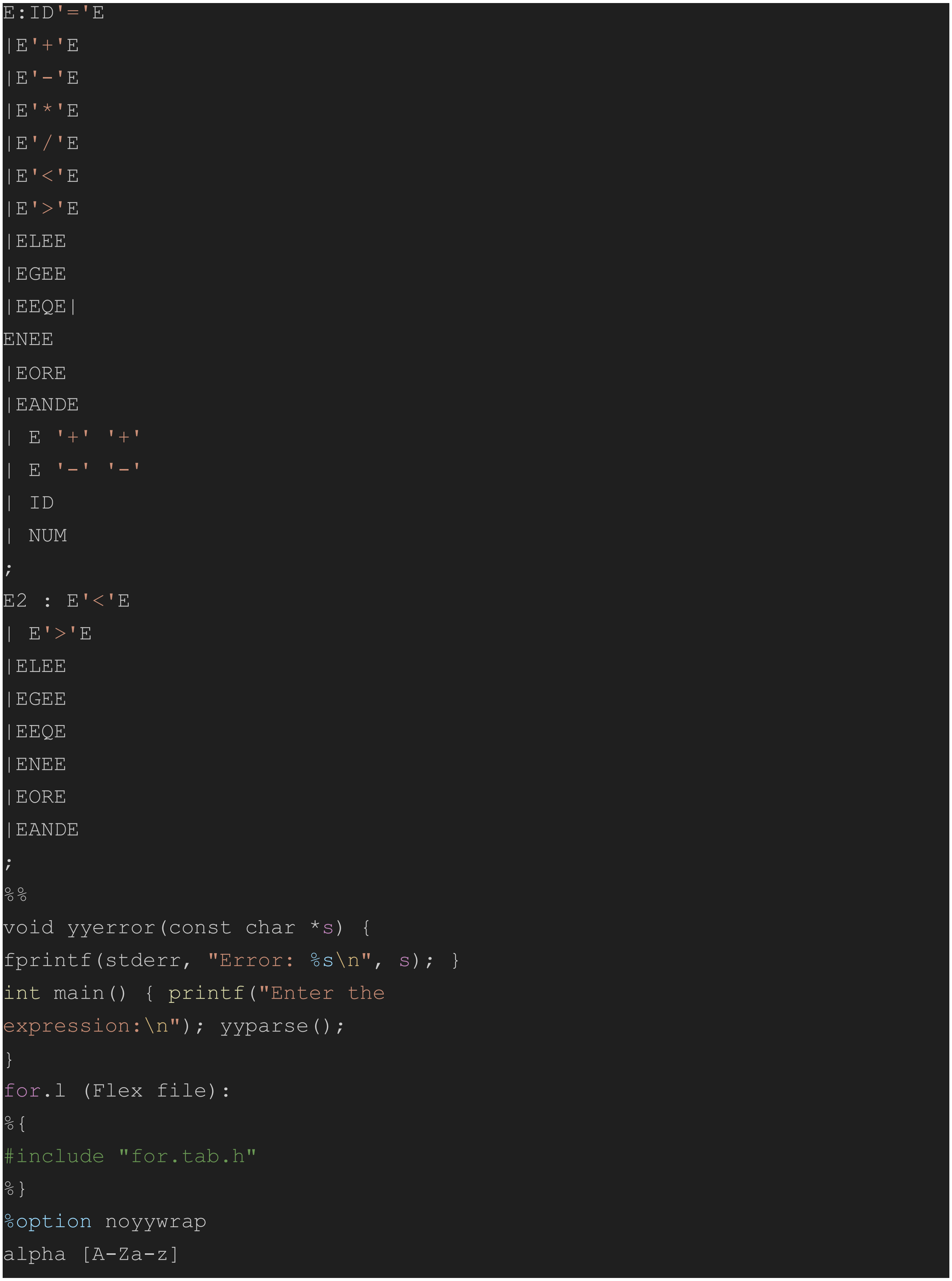


```
Enter the expression:
for(i=0;i<n;i++)
i=i+1;
Input accepted
```

Program

YACC

```
%{
#include <stdio.h>
#include <stdlib.h>
int yylex(void); void
yyerror(const char *s);
%}
%token ID NUM FOR LE GE EQ NE OR AND
%right "="
%left OR AND
%left '>' '<' LE GE EQ NE
%left '+' '-'
%left '*' '/'
%right UMINUS
%left '!'
%%
S : ST {printf("Input accepted\n"); exit(0);}
ST : FOR '(' E ';' E2 ';' E ')' DEF
;
DEF : '{' BODY '}'
| E';'
| ST
|
;
BODY : BODY BODY
|E';'
| ST
|
;
```



```
E:ID'='E
|E'+'E
|E'-'E
|E'*'E
|E'/'E
|E'<'E
|E'>'E
|ELEE
|EGEE
|EEQE|
ENEE
|EORE
|EANDE
| E '+' '+'
| E '-' '-'
| ID
| NUM
;
E2 : E'<'E
| E'>'E
|ELEE
|EGEE
|EEQE
|ENEE
|EORE
|EANDE
;
%%
void yyerror(const char *s) {
fprintf(stderr, "Error: %s\n", s); }
int main() { printf("Enter the
expression:\n"); yyparse();
}
for.l (Flex file):
%{
#include "for.tab.h"
%}
%option noyywrap
alpha [A-Za-z]
```

LEX

```
digit [0-9]
%%
[\t \n] for
return FOR;
{digit}+ return NUM; {alpha}({alpha}|{digit})* return ID;
"<=" return LE;
">=" return GE;
"==" return EQ;
"!=" return NE;
"||" return OR;
"&&" return AND;
. return yytext[0];
%%
```

Writeup
This program accepts for loop as input using LEX and YACC