%option noyywrap

preprocessor #.\*

keywords "auto"|"double"|"struct"|"break"|"else"|"long"|"switch"|"case"|"enum"|"register"|"typedef"|"extern"|"return"|"union"|"const"|"short"|"unsigned"|"continue"|"signed"|"void"|"default"|"goto"|"sizeof"|"volatile"|"do"|"static"|"while"|"for"|"int"|"if"|"char"|"float"

id [\_a-zA-Z][\_a-zA-Z0-9]\*

ints [0-9]+

arithOp [+\-\*/%]|"++"|"--"

logicalOp "&&"|"||"|"!"

assignmentOp "="|"+="|"-="|"/="|"\*="

relOp "=="|"<="|">="|"<"|">"|"!="

bitwiseOp [&|^-]|"<<"|">>"

singleComment "//".\*

multiComment "/\*"(.|\n)\*"\*/"

string \".\*\"

punctuator [;\[\](\){\}]

%%

{preprocessor} {printf("Preprocessor directive: %s\n", yytext);}

{keywords} {printf("Keyword: %s\n", yytext);}

{id} {printf("Identifier: %s\n", yytext);}

{ints} {printf("Integers: %s\n", yytext);}

{arithOp} {printf("Arithmetic Op: %s\n", yytext);}

{logicalOp} {printf("Logical Op: %s\n", yytext);}

{assignmentOp} {printf("Assignment Op: %s\n", yytext);}

{relOp} {printf("Relational Op: %s\n", yytext);}

{bitwiseOp} {printf("Bitwise Op: %s\n", yytext);}

{singleComment} {printf("Single comment: %s\n", yytext);}

{multiComment} {printf("Multi comment: %s\n", yytext);}

{string} {printf("String Literal: %s\n", yytext);}

{punctuator} {printf("Punctuator: %s\n", yytext);}

%%

int main()

{

yyin = fopen("in.c", "r");

yylex();

return 0;

}