Laboratory 8 - FLEX

Pop Ruxandra Paula

* Github: <https://github.com/ruxipaula/flcd>

%{  
#include <stdio.h>  
#include <string.h>  
int lines = 0;  
%}  
  
%option noyywrap  
  
STRING \"[a-zA-Z0-9]\*\"  
INTEGER [+-]?[1-9][0-9]\*|0  
CONST {STRING}|{INTEGER}  
IDENTIFIER [a-zA-Z][a-zA-Z0-9]{0,5}  
  
%%  
  
start {printf("Reserved word: %s\n", yytext);}  
array {printf( "Reserved word: %s\n", yytext);}  
if {printf( "Reserved word: %s\n", yytext);}  
else {printf( "Reserved word: %s\n", yytext);}  
readInt {printf( "Reserved word: %s\n", yytext);}  
readString {printf( "Reserved word: %s\n", yytext);}  
while {printf( "Reserved word: %s\n", yytext);}  
print {printf( "Reserved word: %s\n", yytext);}  
int {printf( "Reserved word: %s\n", yytext);}  
string {printf( "Reserved word: %s\n", yytext);}  
do {printf( "Reserved word: %s\n", yytext);}  
for {printf( "Reserved word: %s\n", yytext);}  
return {printf( "Reserved word: %s\n", yytext);}  
  
{IDENTIFIER} {printf( "Identifier: %s\n", yytext );}  
{CONST} {printf( "Constant: %s\n", yytext );}  
  
":" {printf( "Separator: %s\n", yytext );}  
";" {printf( "Separator: %s\n", yytext );}  
"," {printf( "Separator: %s\n", yytext );}  
"." {printf( "Separator: %s\n", yytext );}  
"{" {printf( "Separator: %s\n", yytext );}  
"}" {printf( "Separator: %s\n", yytext );}  
"(" {printf( "Separator: %s\n", yytext );}  
")" {printf( "Separator: %s\n", yytext );}  
"[" {printf( "Separator: %s\n", yytext );}  
"]" {printf( "Separator: %s\n", yytext );}  
"+" {printf( "Operator: %s\n", yytext );}  
"-" {printf( "Operator: %s\n", yytext );}  
"\*" {printf( "Operator: %s\n", yytext );}  
"/" {printf( "Operator: %s\n", yytext );}  
"<" {printf( "Operator: %s\n", yytext );}  
">" {printf( "Operator: %s\n", yytext );}  
"<=" {printf( "Operator: %s\n", yytext );}  
">=" {printf( "Operator: %s\n", yytext );}  
"!=" {printf( "Operator: %s\n", yytext );}  
"==" {printf( "Operator: %s\n", yytext );}  
"&&" {printf( "Operator: %s\n", yytext );}  
"=" {printf( "Separator: %s\n", yytext );}  
"!" {printf( "Operator: %s\n", yytext );}  
  
  
[ \t]+ {}  
[\n]+ {lines++;}  
  
[+-]?0[0-9]\* {printf("Error - illegal constant at line %d\n", lines);}  
  
[a-zA-Z][a-zA-Z0-9]{6,} {printf("Error - illegal size of the identifier at line %d\n", lines);}  
  
[0-9~@#$%^][a-zA-Z0-9]{0,5} {printf("Error - illegal identifier at line %d\n", lines);}  
  
%%  
void main(argc, argv)  
int argc;  
char\*\* argv; {   
if (argc > 1)  
{  
 FILE \*file;  
 file = fopen(argv[1], "r");  
 if (!file)  
 {  
 fprintf(stderr, "Could not open %s\n", argv[1]);  
 exit(1);  
 }  
 yyin = file;  
}  
  
yylex();  
printf("Correct program!");  
}

P1.in

int a;  
int b;  
int c;  
a=2;  
b=3;  
c=10;  
if(a>=b&&a>=c){  
return a;  
}  
if(b>=a&&b>=c){  
return b;  
}  
if(c>=a&&c>=b){  
return c;  
}