LEX Specification and Demonstration

LEX Specification (lang.lxi)

```
%{
#include <stdio.h>
%}

%%
"//".*                    { /* Ignore single line comments */ }
"var"                     { printf("KEYWORD(var) "); }
"while"                   { printf("KEYWORD(while) "); }
"if"                      { printf("KEYWORD(if) "); }
"else"                    { printf("KEYWORD(else) "); }
"print"                   { printf("KEYWORD(print) "); }
"readInt"                 { printf("FUNCTION(readInt) "); }
[0-9]+                    { printf("NUMBER(%s) ", yytext); }
[a-zA-Z_][a-zA-Z0-9_]*    { printf("IDENTIFIER(%s) ", yytext); }
"+"|"-"|"*"|"/"|"%"       { printf("OPERATOR(%s) ", yytext); }
"=="|"!="|"<"|"<="|">"|">="   { printf("COMPARISON(%s) ", yytext); }
"="                       { printf("ASSIGNMENT(=) "); }
";"                       { printf("SEMICOLON(;) "); }
"{"                       { printf("LEFT_BRACE({) "); }
"}"                       { printf("RIGHT_BRACE(}) "); }
"("                       { printf("LEFT_PAREN(() "); }
")"                       { printf("RIGHT_PAREN()) "); }
[
```

LEX Specification and Demonstration

```
]+                  { /* Ignore whitespace */ }
.                       { printf("UNKNOWN(%s) ", yytext); }
%%

int main(int argc, char **argv) {
    if (argc > 1) {
        FILE *file;
        file = fopen(argv[1], "r");
        if (!file) {
            fprintf(stderr, "Could not open %s
", argv[1]);
            return 1;
        }
        yyin = file;
    }

    yylex();
    return 0;
}
```

Demonstration of Scanner Output

Demonstration of Scanner Output:

For p1.aks (Verify if a number is prime):

LEX Specification and Demonstration

Expected Output:

IDENTIFIER(n) ASSIGNMENT(=) FUNCTION(readInt) SEMICOLON(;) IDENTIFIER(prime) ASSIGNMENT(=) NUMBER(1) SEMICOLON(;) ...

For p2.aks (Compute the maximum of 3 numbers):

Expected Output:

IDENTIFIER(x) COMMA IDENTIFIER(y) COMMA IDENTIFIER(z) SEMICOLON(;) IDENTIFIER(x) ASSIGNMENT(=) FUNCTION(readInt) SEMICOLON(;) ...

For p3.aks (Compute the sum of n numbers):

Expected Output:

IDENTIFIER(n) ASSIGNMENT(=) FUNCTION(readInt) SEMICOLON(;) IDENTIFIER(sum) ASSIGNMENT(=) NUMBER(0) SEMICOLON(;) ...