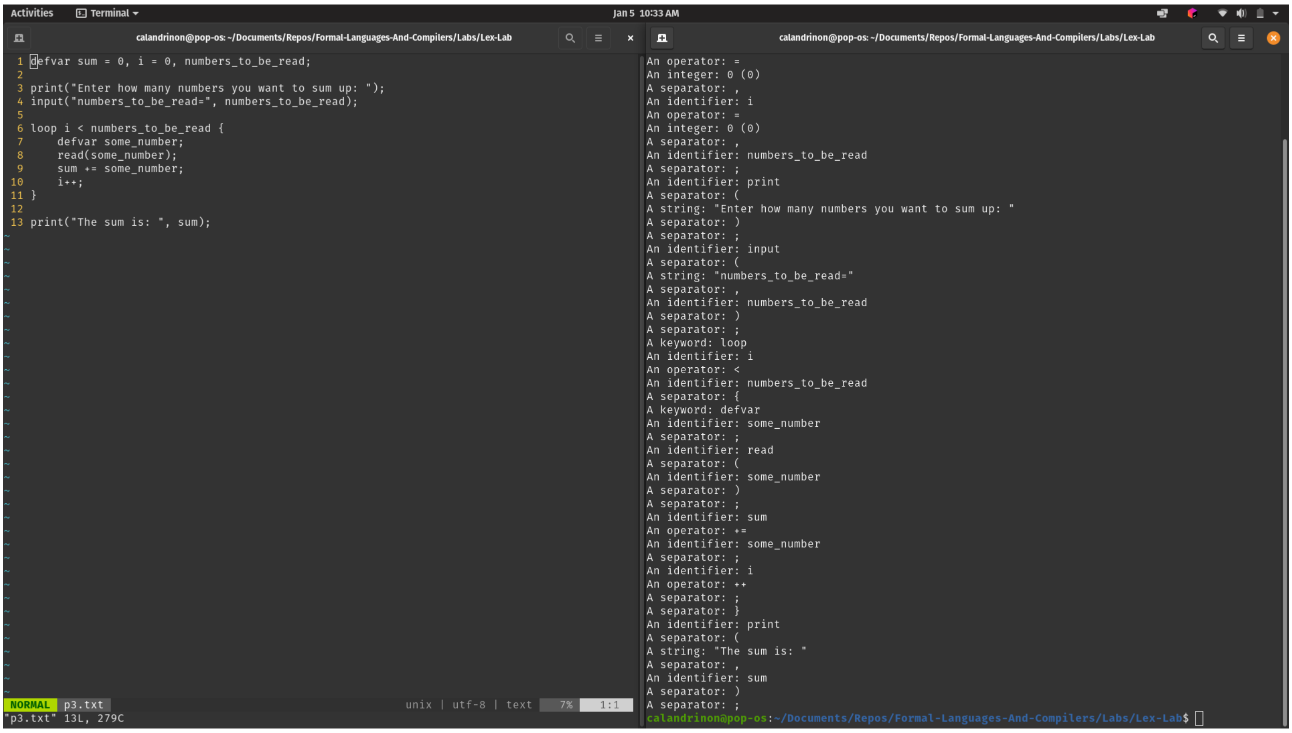
Lab 8 – Lex

Specification:

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

DIGIT                [0-9]
ID           [a-z][_a-z0-9]*
STRING       \".*\"

%%

{DIGIT}+     printf( "An integer: %s (%d)\n", yytext, atoi( yytext ) );

{DIGIT}+"."{DIGIT}*  printf( "A float: %s (%g)\n", yytext, atof( yytext ) );
"#".* printf("A comment: \"%s\"\n", yytext);

"defvar"|"deflist"|"if"|"else"|"else if"|"and"|"or"|"not"|"loop" printf( "A keyword: %s\n", yytext );


{ID}         printf( "An identifier: %s\n", yytext );

{STRING} printf("A string: %s\n", yytext);

"+"|"-"|"*"|"/"|"%"|"="|"<"|">"|"<="|">="|"+="|"-="|"++"|"--"|"**" printf( "An operator: %s\n",
yytext );

";"|"["|"]"|"{"|"}"|"("|")"|","|":" printf("A separator: %s\n", yytext);

"{"[^}\n]*"}"        /* eat up one-line comments */

[ \t\n]+             /* eat up whitespace */

. printf("Eroare: %s\n", yytext);
%%
main( argc, argv )
int argc;
char **argv;
{
   ++argv, --argc; /* skip over program name */
   if ( argc > 0 )
   yyin = fopen( argv[0], "r" );
   else
    yyin = stdin;
   yylex();
}
```

Demo:



Activities
Terminal
Jan 5 10:33 AM
calandrinon@pop-os: ~/Documents/Repos/Formal-Languages-And-Compilers/Labs/Lex-Lab
calandrinon@pop-os: ~/Documents/Repos/Formal-Languages-And-Compilers/Labs/Lex-Lab

```
defvar sum = 0, i = 0, numbers_to_be_read;

print("Enter how many numbers you want to sum up: ");
input("numbers_to_be_read=", numbers_to_be_read);

loop i < numbers_to_be_read {
    defvar some_number;
    read(some_number);
    sum += some_number;
    i++;
}

print("The sum is: ", sum);
```

```
NORMAL  p3.txt                                          unix | utf-8 | text   7%    1:1
"p3.txt" 13L, 279C
```

```
An operator: =
An integer: 0 (0)
A separator: ,
An identifier: i
An operator: =
An integer: 0 (0)
A separator: ,
An identifier: numbers_to_be_read
A separator: ;
An identifier: print
A separator: (
A string: "Enter how many numbers you want to sum up: "
A separator: )
A separator: ;
An identifier: input
A separator: (
A string: "numbers_to_be_read="
A separator: ,
An identifier: numbers_to_be_read
A separator: )
A separator: ;
A keyword: loop
An identifier: i
An operator: <
An identifier: numbers_to_be_read
A separator: {
A keyword: defvar
An identifier: some_number
A separator: ;
An identifier: read
A separator: (
An identifier: some_number
A separator: )
A separator: ;
An identifier: sum
An operator: +=
An identifier: some_number
A separator: ;
An identifier: i
An operator: ++
A separator: ;
A separator: }
An identifier: print
A separator: (
A string: "The sum is: "
A separator: ,
An identifier: sum
A separator: )
A separator: ;
calandrinon@pop-os:~/Documents/Repos/Formal-Languages-And-Compilers/Labs/Lex-Lab$ 
```