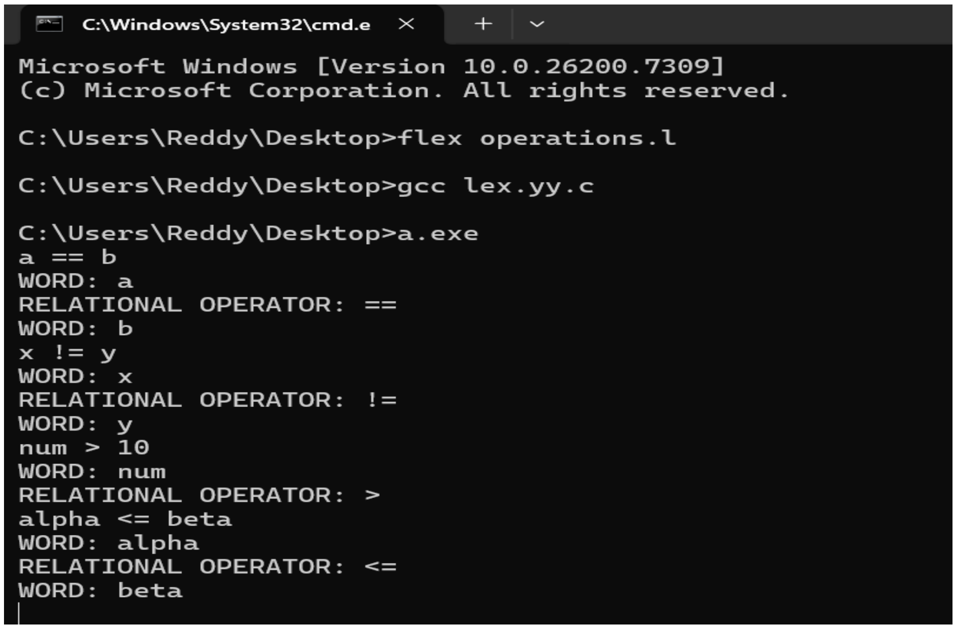
Exp 40: -Relational Operator

Aim: To identify relational operator and words using lexical programming language.

Code:

```
%{
#include <stdio.h>
%}
%option noyywrap
%%
"=="       { printf("RELATIONAL OPERATOR: ==\n"); }
"!="       { printf("RELATIONAL OPERATOR: !=\n"); }
">="       { printf("RELATIONAL OPERATOR: >=\n"); }
"<="       { printf("RELATIONAL OPERATOR: <=\n"); }
">"        { printf("RELATIONAL OPERATOR: >\n"); }
"<"        { printf("RELATIONAL OPERATOR: <\n"); }
[a-zA-Z]+   { printf("WORD: %s\n", yytext); }
.|\n       { /* ignore */ }
%%
int main() {
   yylex();
   return 0;
}
```

Output:



```
Microsoft Windows [Version 10.0.26200.7309]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Reddy\Desktop>flex operations.l

C:\Users\Reddy\Desktop>gcc lex.yy.c

C:\Users\Reddy\Desktop>a.exe
a == b
WORD: a
RELATIONAL OPERATOR: ==
WORD: b
x != y
WORD: x
RELATIONAL OPERATOR: !=
WORD: y
num > 10
WORD: num
RELATIONAL OPERATOR: >
alpha <= beta
WORD: alpha
RELATIONAL OPERATOR: <=
WORD: beta
```