Compiler Design 19CSE401

Experiment 1E

Name : Anush E

Roll No: CH.EN.U4CSE22005

Aim:

To write a Lex program that identifies relational operators (==, !=, <, <=, >, >=) from the input.

Procedure:

- Open a terminal in your Linux environment (e.g., Ubuntu).
- Create a new Lex source file using a text editor like gedit, nano, or vim.
  gedit relational_ops.l
- In the Lex file, define the patterns for relational operators using regular expressions.
- In the %% section, match each operator (==, !=, <=, >=, <, >) and print a descriptive message when matched.

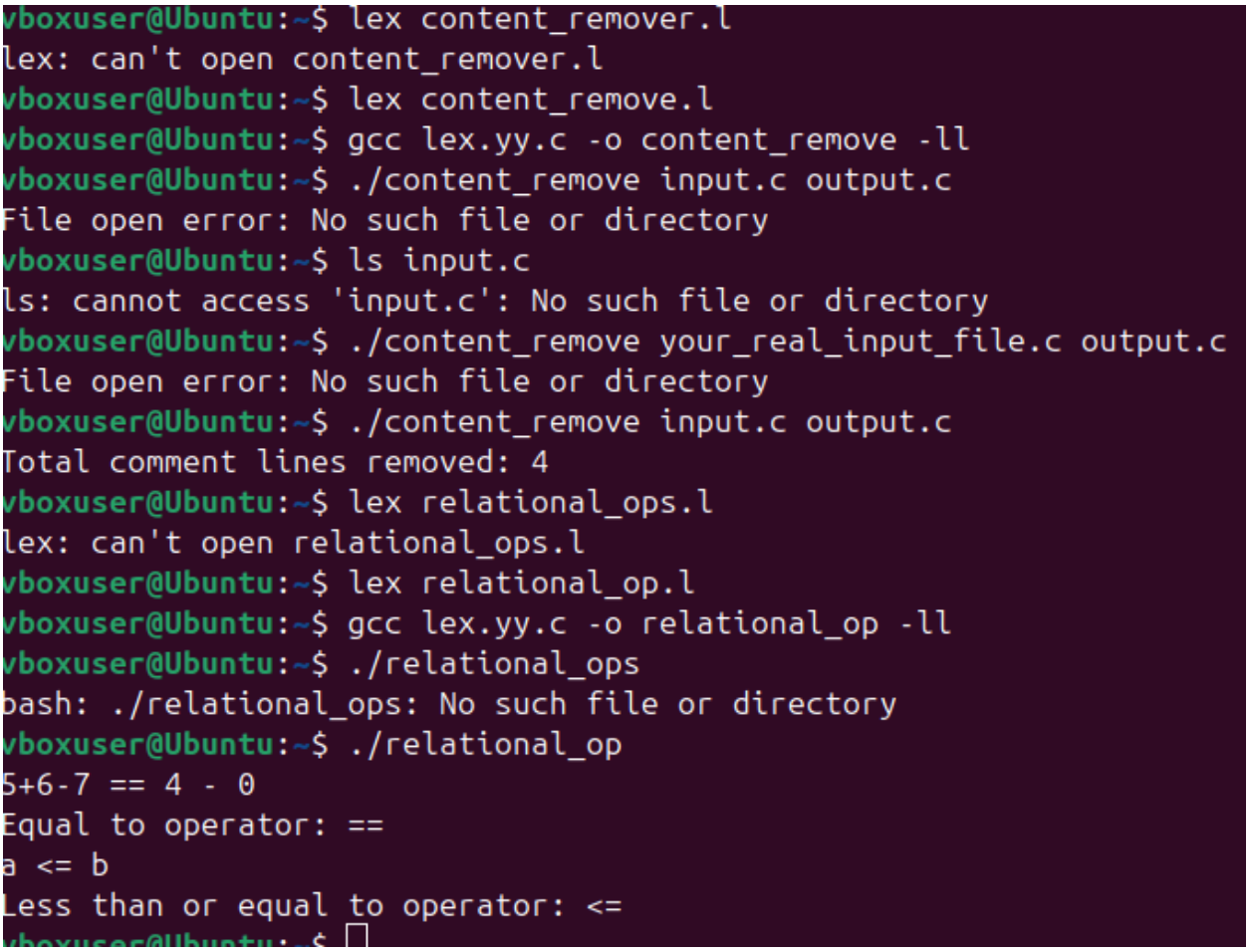
- Define a main() function after the second %% that calls yylex() to start the lexical analysis.
- Compile the Lex program using lex and gcc.
- Run the executable program.
- Type or paste an expression with relational operators (e.g., a <= b && a != b) and press Ctrl + D to end input (EOF).
- Observe the output showing which relational operators were detected and their descriptions.

Program Code:

```
%%

"==" {

printf("Equal to operator: %s\n", yytext);

}

"!=" {

printf("Not equal to operator: %s\n", yytext); } "<="
{ printf("Less than or equal to operator: %s\n",
yytext); } ">=" {
```

```
printf("Greater than or equal to operator: %s\n", yytext); } "<" {

printf("Less than operator: %s\n", yytext);

} ">" {

printf("Greater than operator: %s\n", yytext);

} .|\n

{ /* Ignore other characters */

}

%%

int main() {

yylex();

return 0;

}
```

Output:



Result:

We had found out the relational operators present in the given expression using lex program.