Experiment III b

**Aim**: Implementation of a calculator using LEX and YACC.

Algorithm

1. Start
2. Read expression as input
3. Check the expression pattern `p` against regular expressions in lex.
   1. If `p` is a number, return ‘num’ after converting from ascii to number.
   2. If `p` is either ‘+’, ‘-’, ‘\*’, ‘/’, ‘(‘, ‘)’ or new line character, return`p`.
4. Check whether the expression satisfies the context free grammar shown below in yacc.

E -> E - E | F

F -> F + F | G

G -> G \* G | H

H -> H / H | I

I -> (I) | J

J -> num

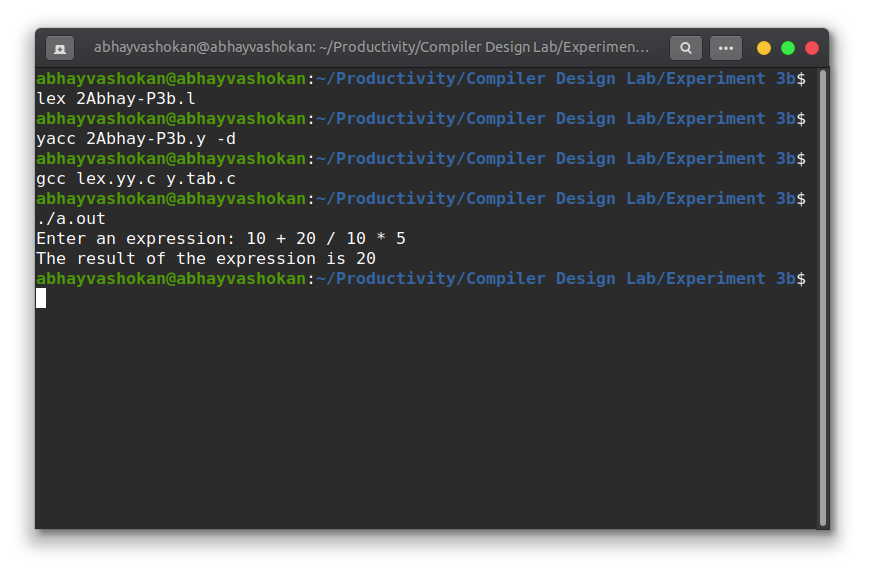
1. If yes, evaluate the expression at each instance and display the result when reduced completely to the start symbol.
2. Else print “Invalid expression”
3. Stop

Output

Enter an expression: 10 + 20 / 10 \* 5

The result of the expression is 20

Screenshot



Readme

1. Compile the lex program using the command

**lex 2Abhay-P3b.l**

2. Compile the yacc program using the command

**lex 2Abhay-P3b.y -d**

3. Now compile and run the **lex.yy.c** and **y.tab.c** files generated using the command

**gcc lex.yy.c y.tab.c && ./a.out**

4. Input an expression

5. The result of the expression shall be obtained as output.

**Result**: Successfully implemented a program to evaluate an expression and display the result.