**CSB 353: Compiler Design**

**LAB 12-13**

**Submitted By:**

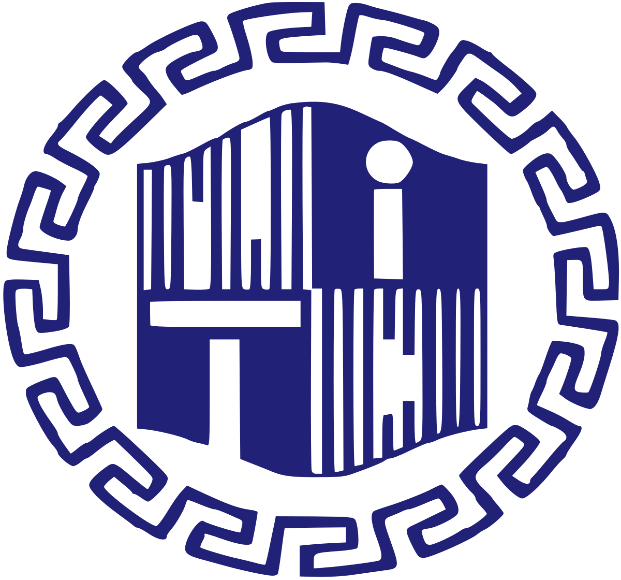
**Name: PREM KUMAR**

**Roll No: 191210037**

**Branch: CSE**

**Semester: 6 th**

**Submitted To: Dr. Shelly Sachdeva**

Department of Computer Science and Engineering

**NATIONAL INSTITUTE OF TECHNOLOGY DELHI**

2019-2023

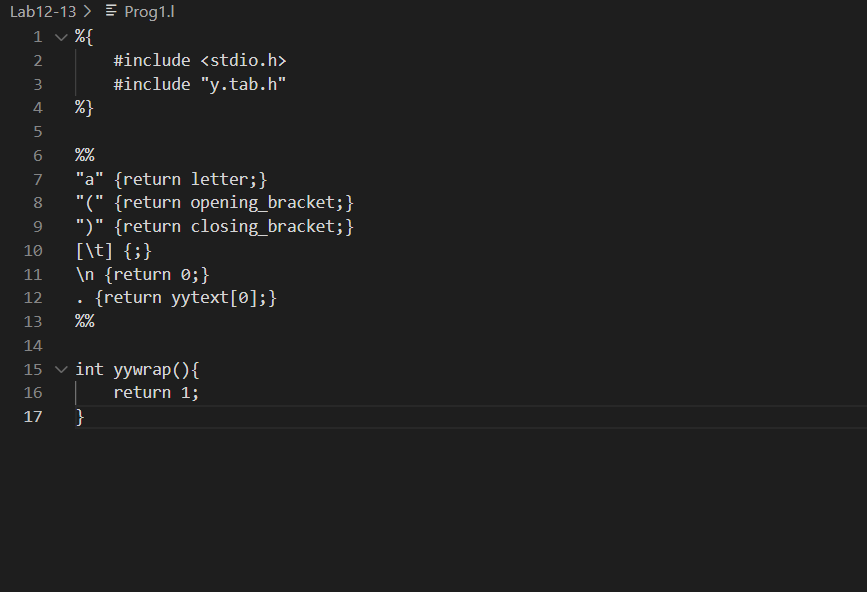
Ques 1. Write a yacc program to check if the parenthesis are balanced and count the

number of matching parenthesis

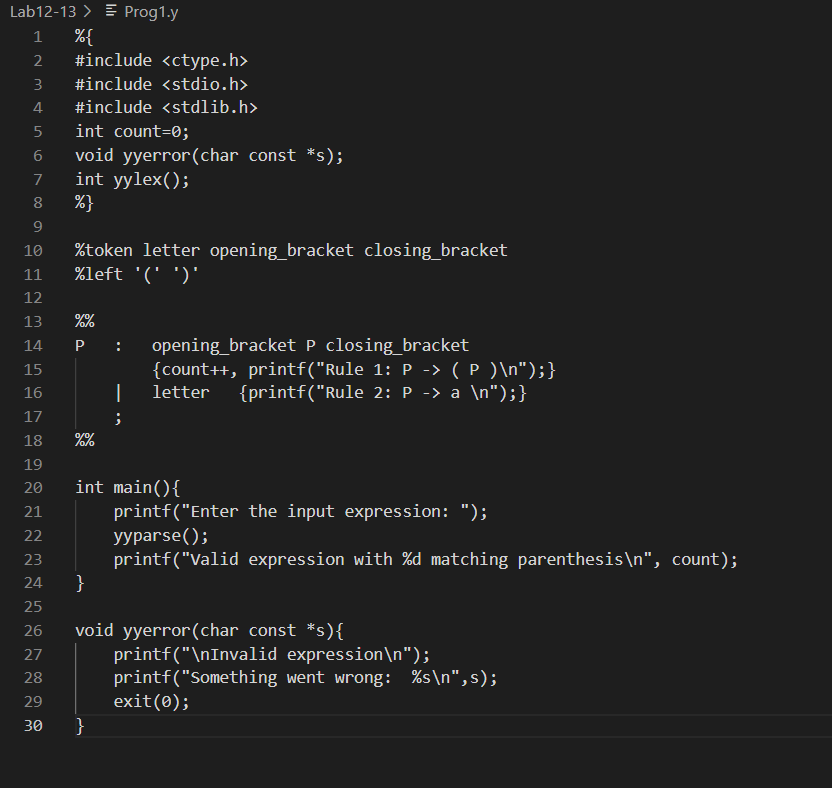
P->(P) | a .

Code:

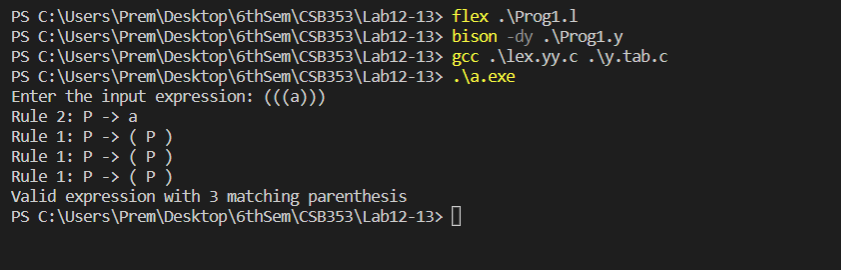
* prog1.l (Lex File)



* prog1.y (Yacc File)



* Output:



Ques 2. Write a YACC program to recognize expressions involving binary numbers as

operands and print the decimal value of the expression

> 1001

Decimal value: 9

>101 + 100

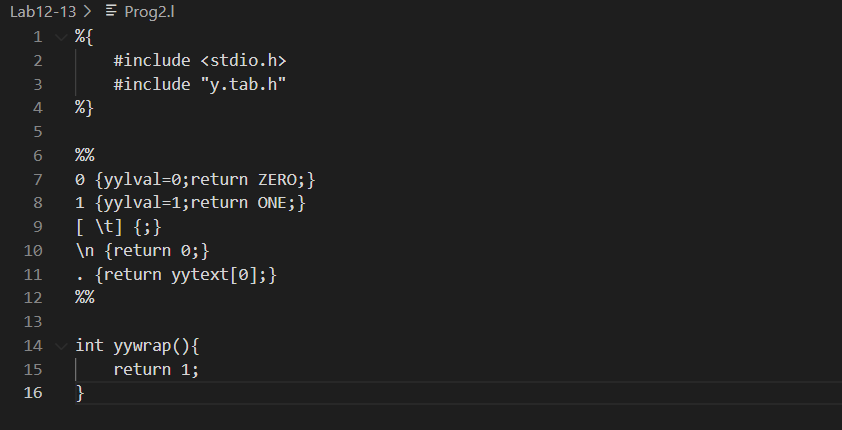
Decimal value: 9

> 201 + 101

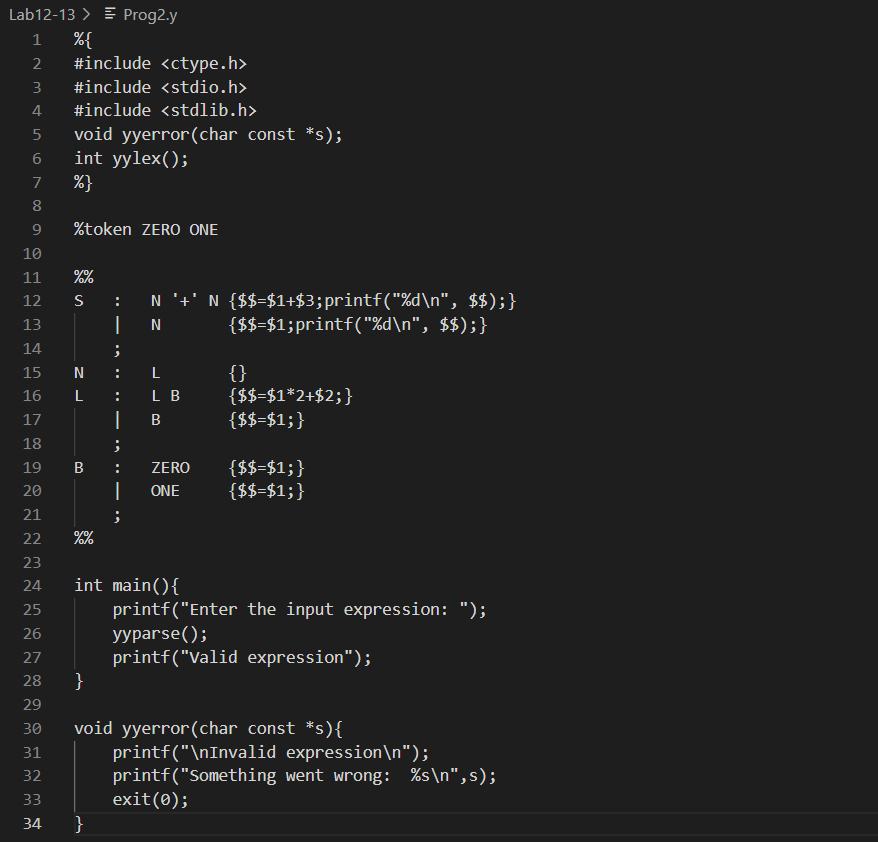
Invalid input.

Code:

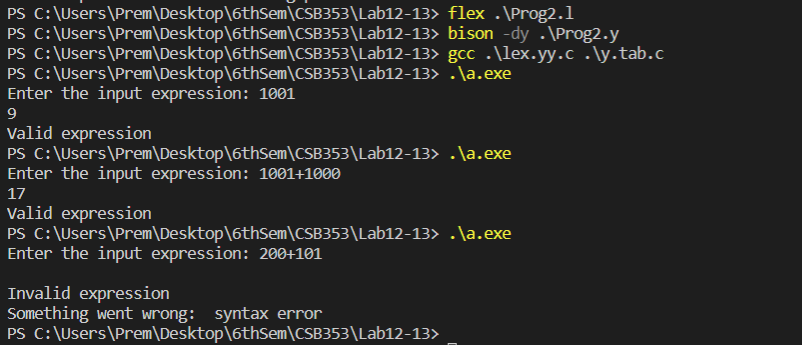
* prog2.l (Lex File)



* prog2.y (Yacc File)



* Output:



Ques 3. Append semantic actions to the grammar of simple desk calculator that performs

simple operations on integer expressions with the grammar:

exp -> exp addop term | term

addop -> + | -

term -> term mulop factor | factor

mulop -> \*

factor -> (exp) | number

number -> number digit | digit

digit -> 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

You are required to write YACC specifications for this grammar so that the

parser evaluates any arithmetic expressions and the output shows each

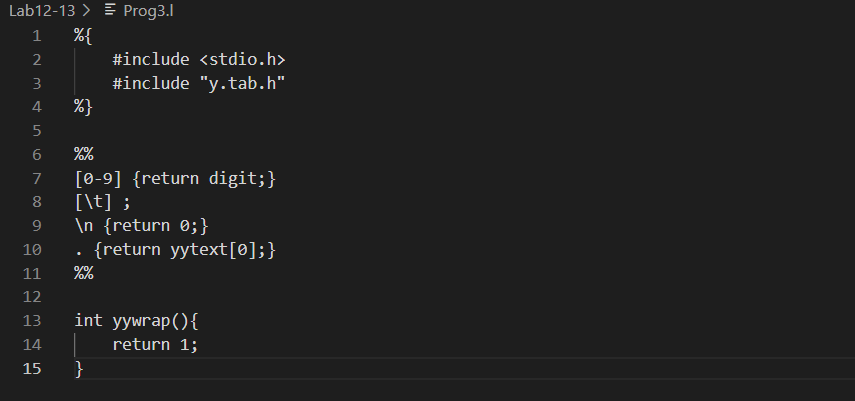
semantic action as it is applied in the parsing process. Show your parsing

sequence for the input string:

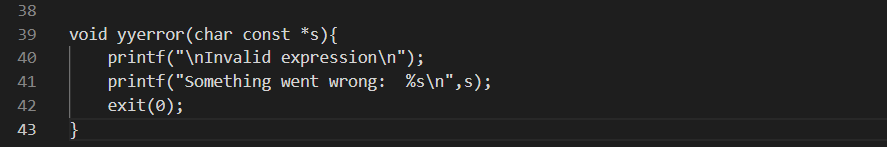
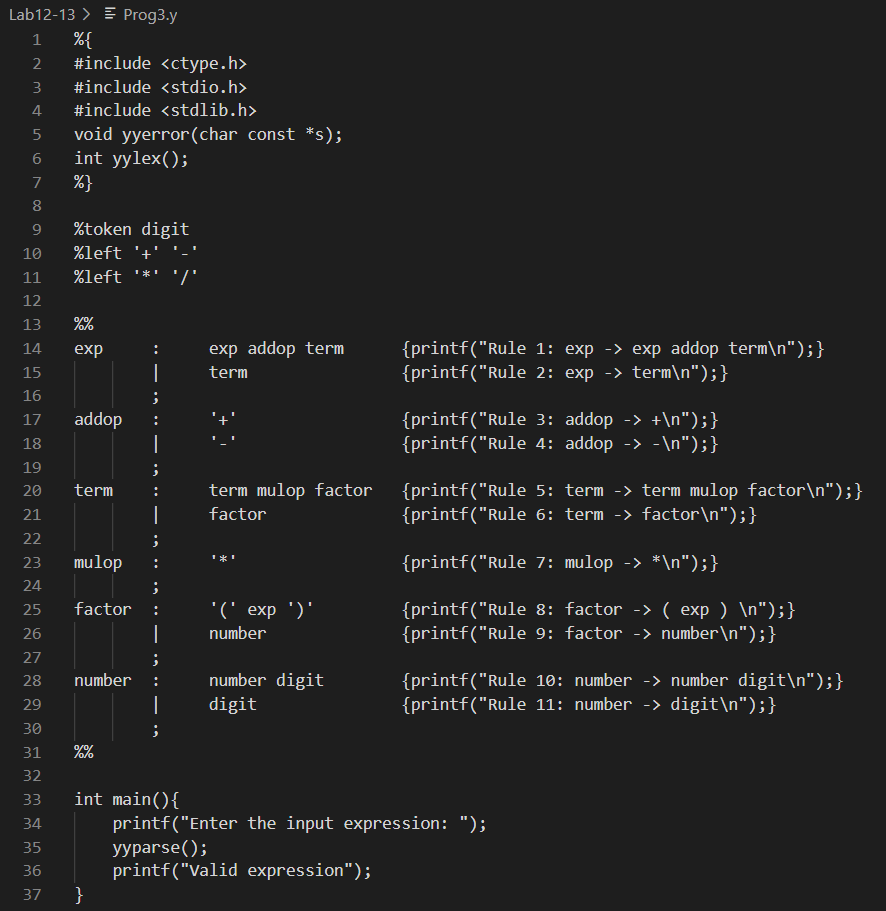
(2+(3\*4)).

Code:

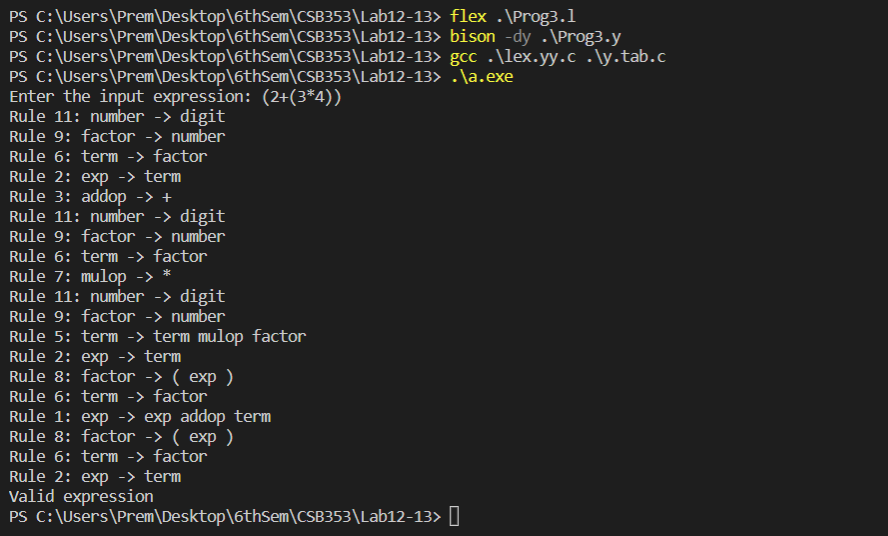
* prog3.l (Lex File)



* prog3.y (Yacc File)



* Output:



Ques 4. Generate a parser for sql select statement (select statement may contain

group by, having clause)and do the type checking wherever required. Count

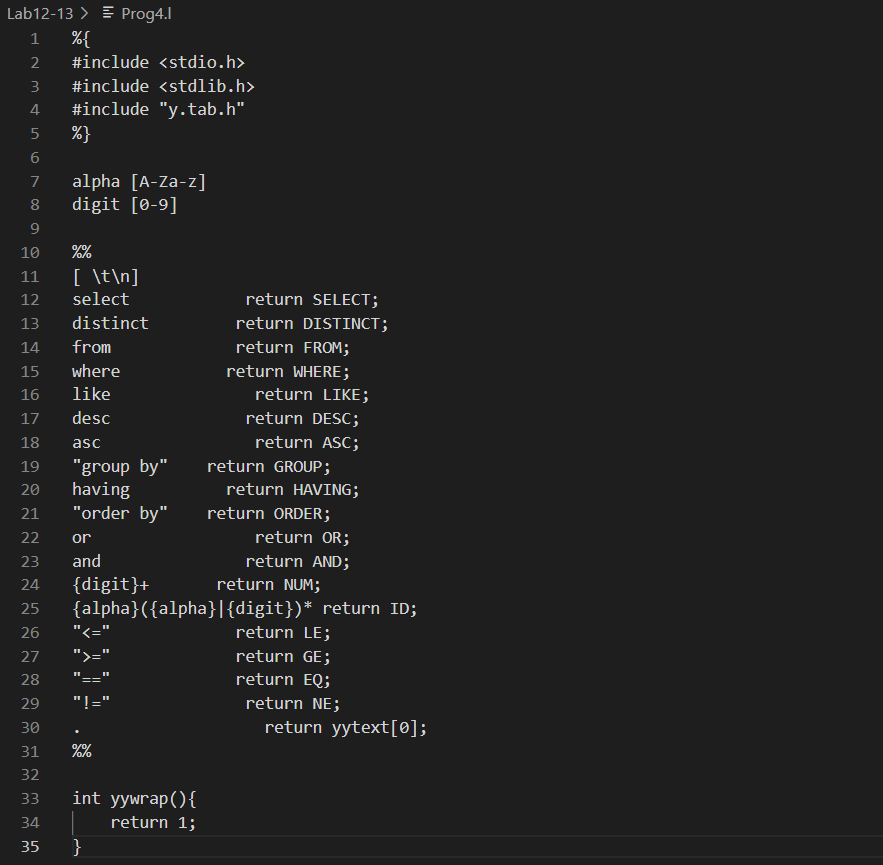
the number of attributes selected and number of attributes used in conditions.

For eg.: select name,address from emp where age>20 group by name having

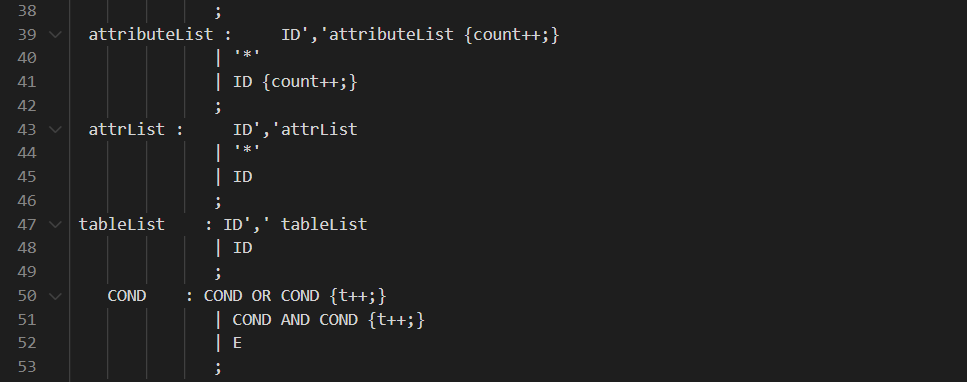
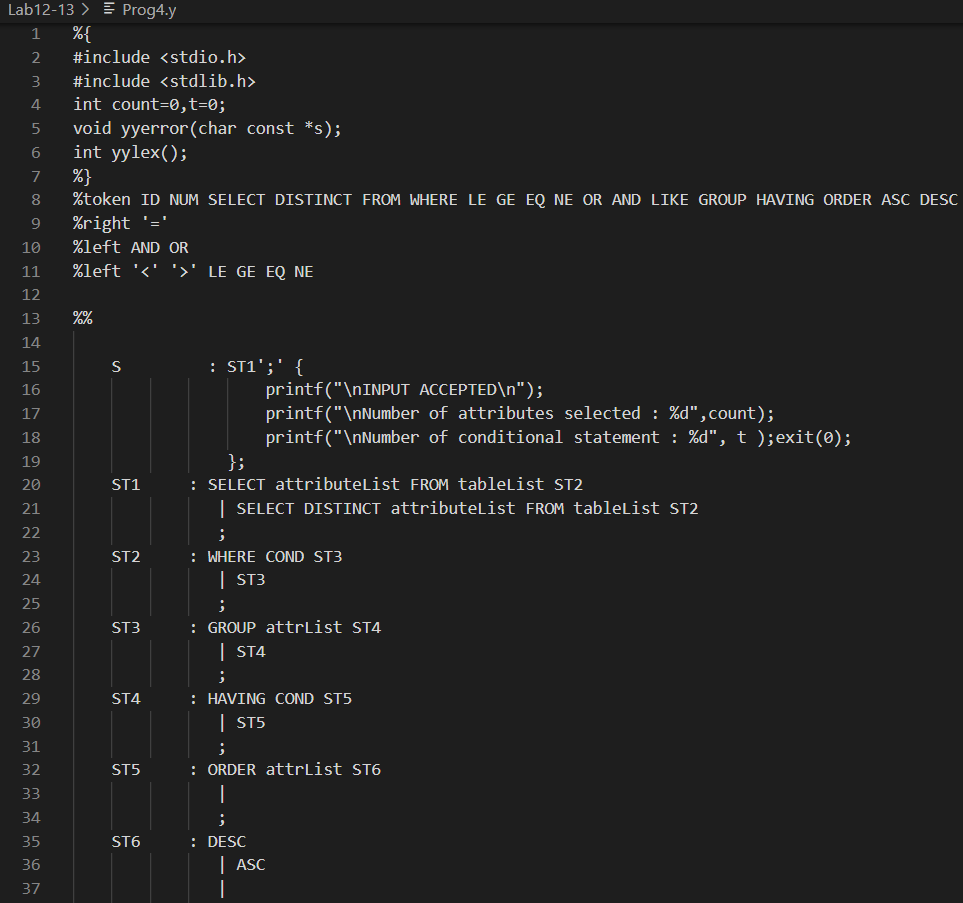
age<40 order by name desc;

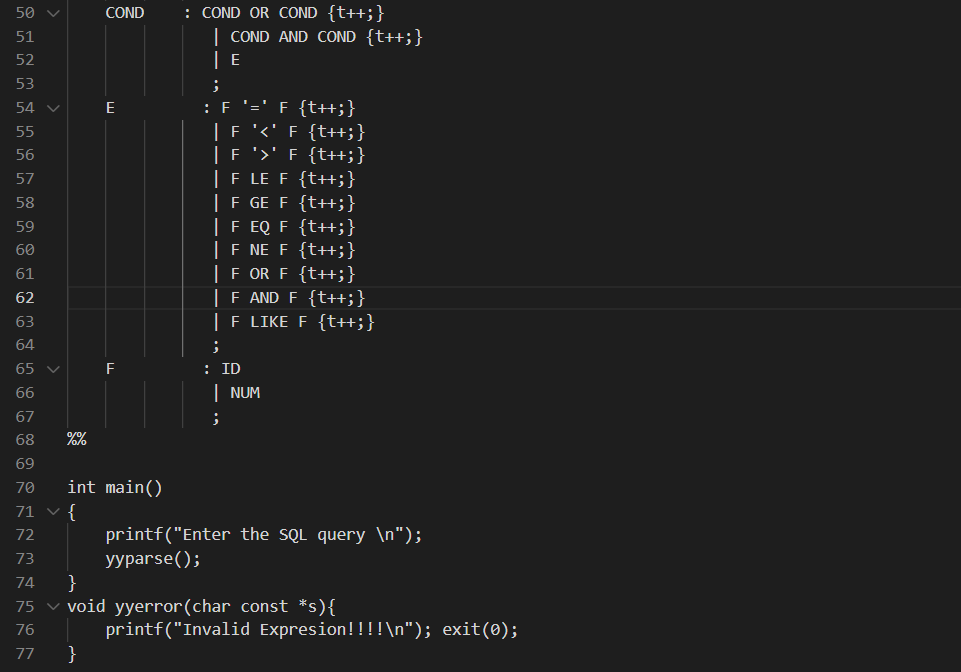
Code:

* prog4.l (Lex File)

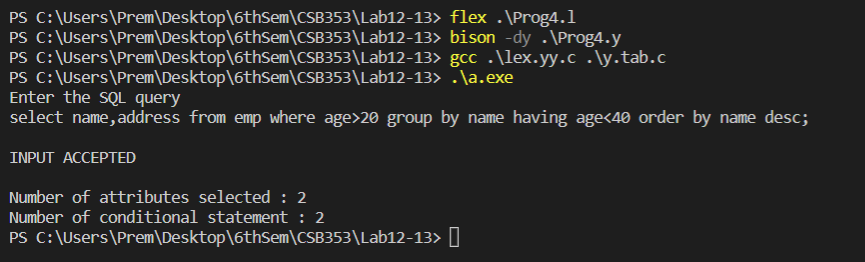


* prog4.y (Yacc File)





* Output:



Ques 5. Write a YACC program to evaluate the following expressions. Specify the

grammar clearly.

Functions Meaning

(+ x1 x2 x3 ….. xn) Calculate Sum of x1, x2 upto xn

(\* x1 x2 x3 ….. xn) Calculate Product of x1, x2 upto xn

(max x1 x2 x3 ….. xn) Calculate Maximum of x1, x2 upto xn

(min x1 x2 x3 ….. xn) Calculate Minimum of x1, x2 upto xn

Sample Input:

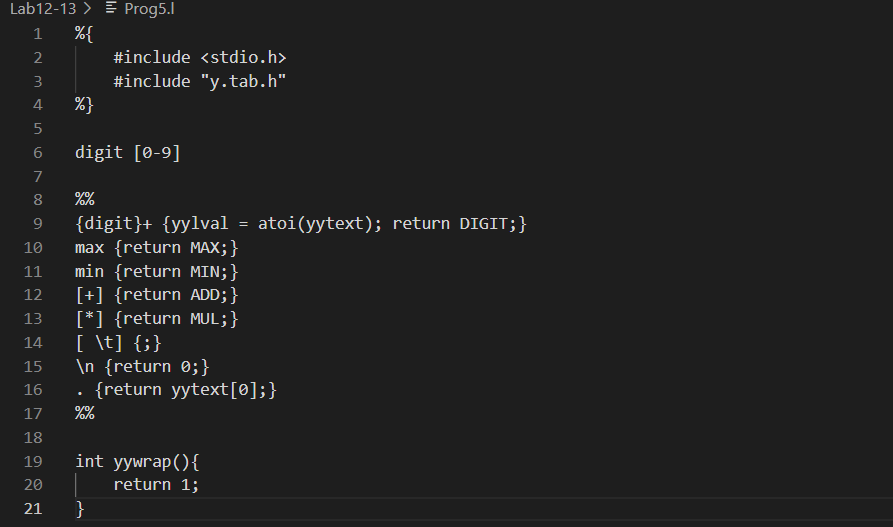
(+ 6 12 18)

Sample Output:

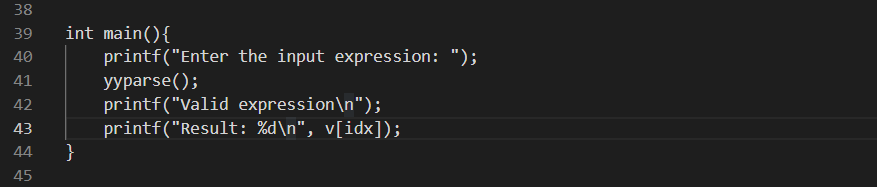
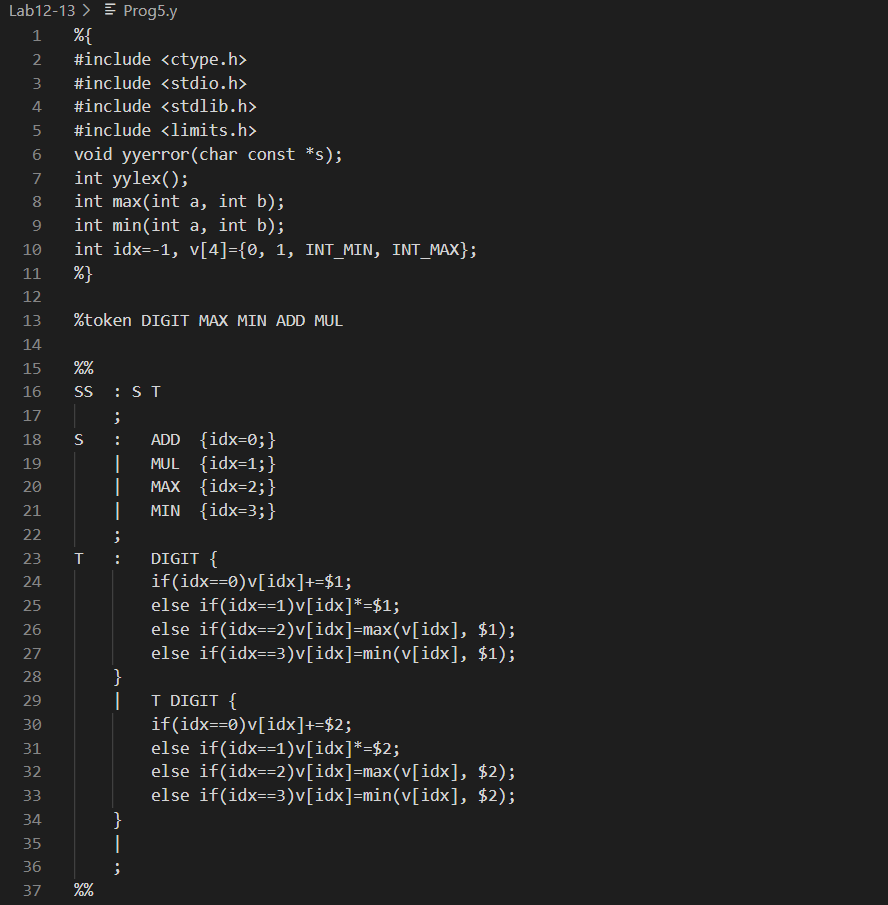
36.

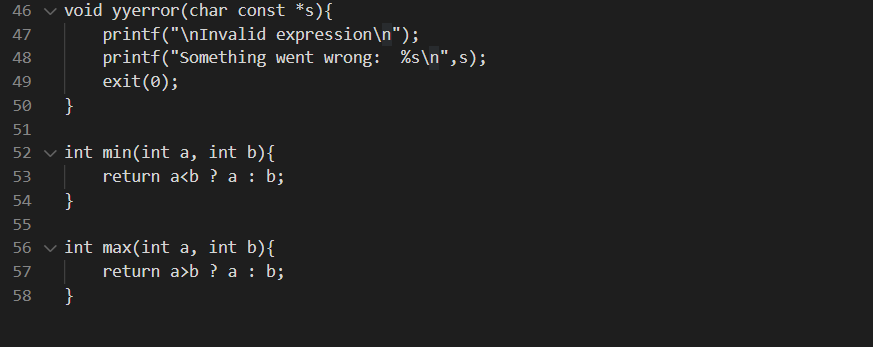
Code:

* prog5.l (Lex File)



* prog5.y (Yacc File)





* Output:

