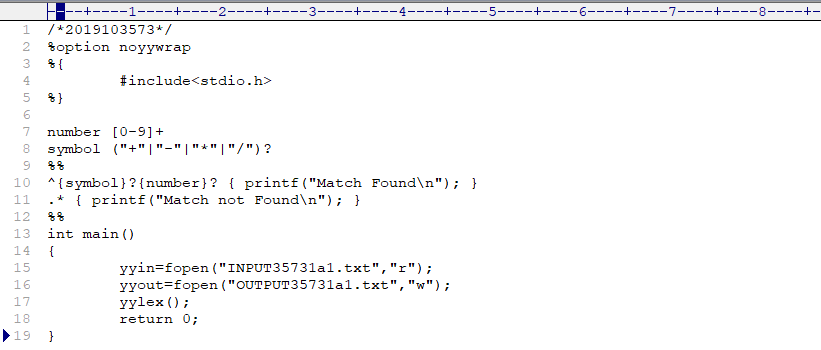
**02-12-2021 2019103573**

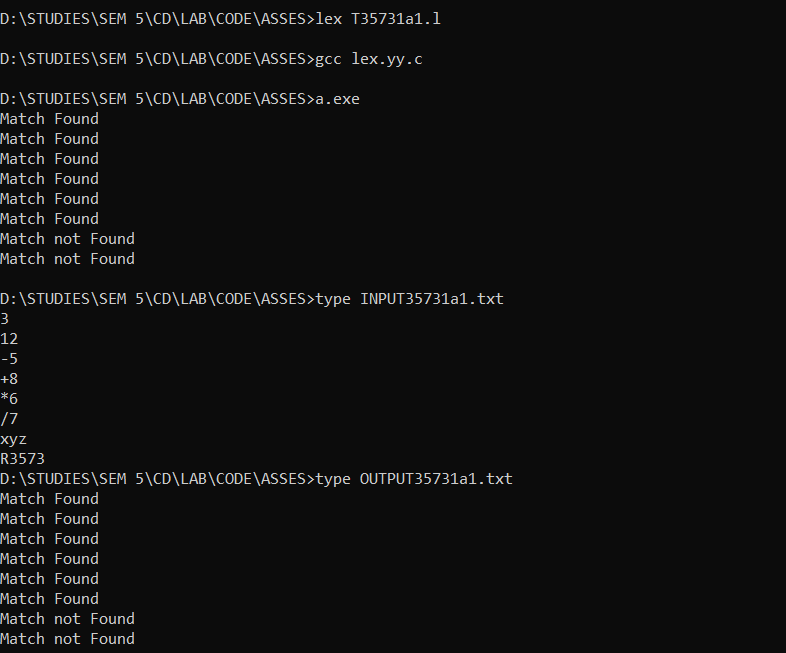
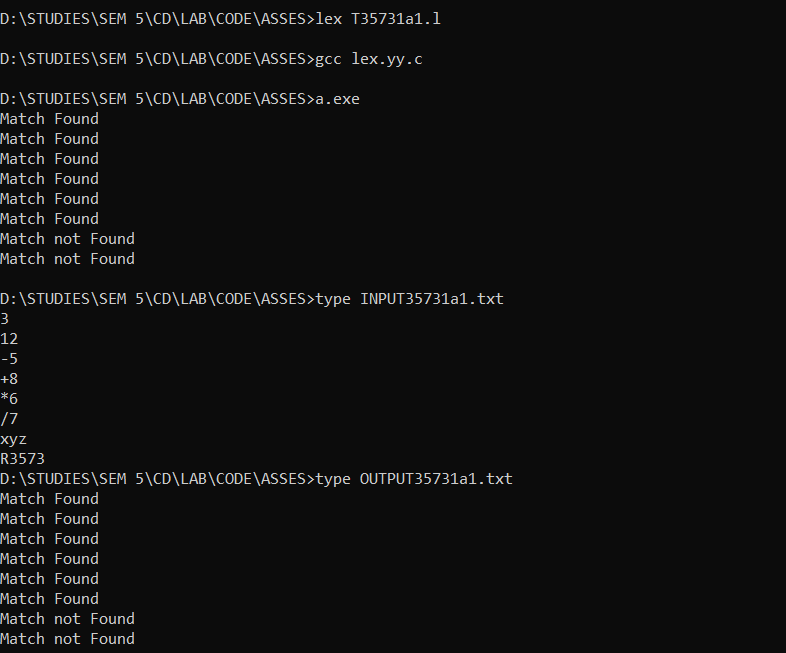
**CS6109 – COMPILER DESIGN LABORATORY**

**LAB ASSESSMENT-SET B**

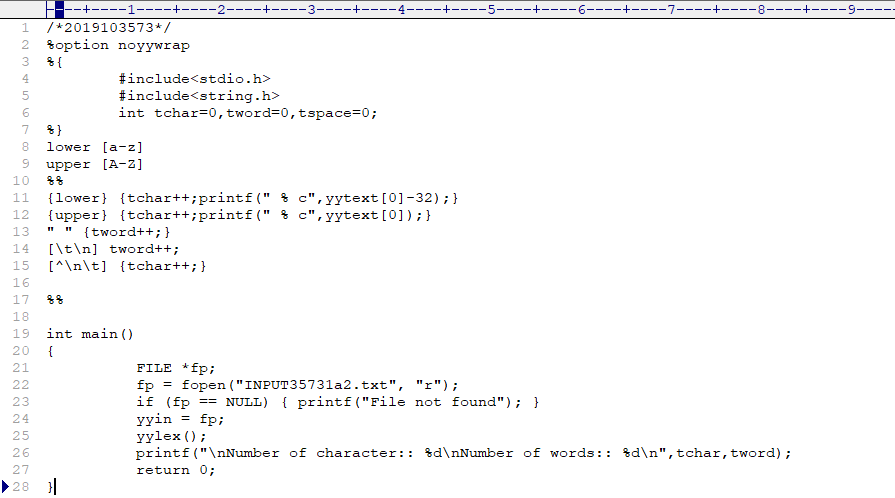
**1.** **a. Write regular definition to display the line of string for the following using LEX.**

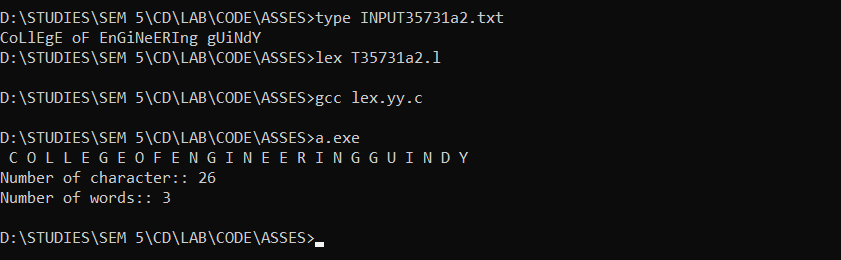
i. Match any string of one or more digits with an optional prefix of +, -, \* and /.

**LEX**

**OUTPUT**

ii. Translating all input string into uppercase, find the character and word count of the input string

**LEX**

**OUTPUT**

iii. Eliminating all C-like comments from a text file

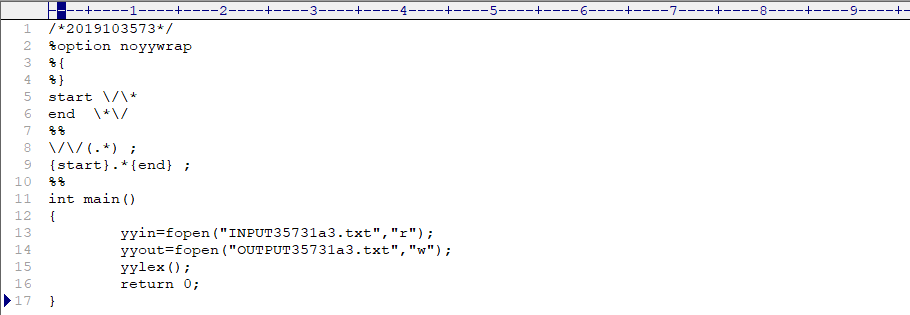
typedef union {

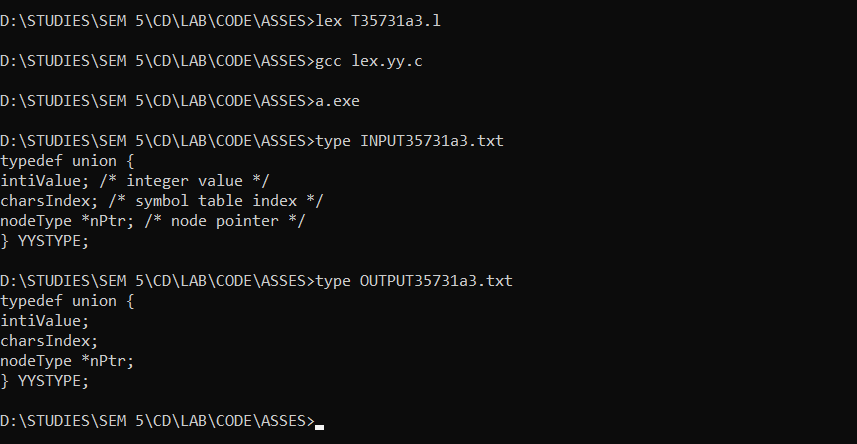
intiValue; /\* integer value \*/

charsIndex; /\* symbol table index \*/

nodeType \*nPtr; /\* node pointer \*/

} YYSTYPE;

**LEX**

**OUTPUT**

**b. Convert the while loop to nested for statement**

int i=1, j=1;

while (i<= 4 || j <= 3)

{

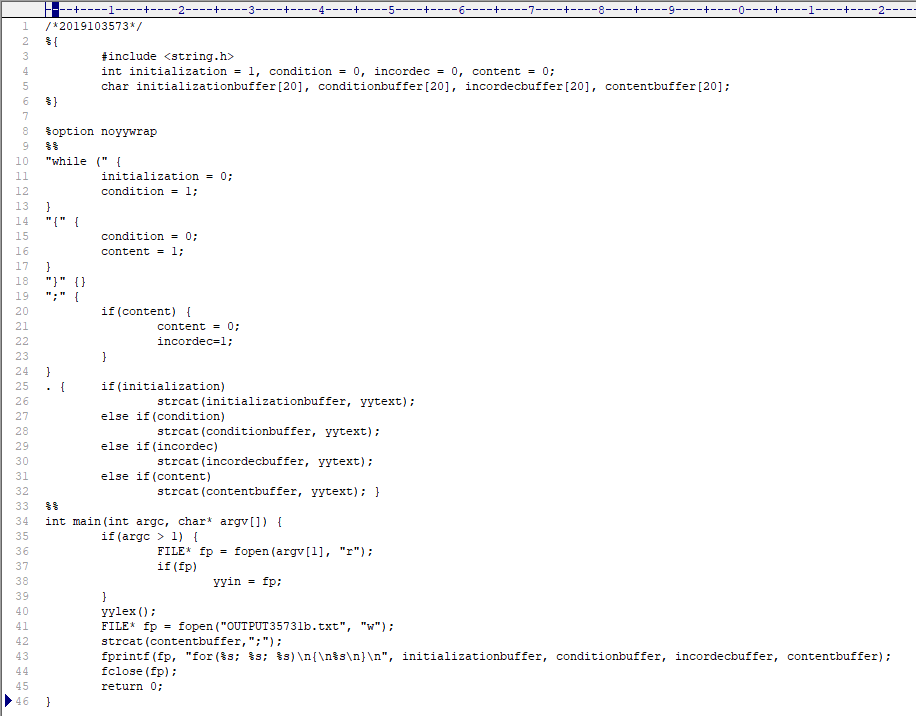
printf("%d %d\n",i, j);

i++;

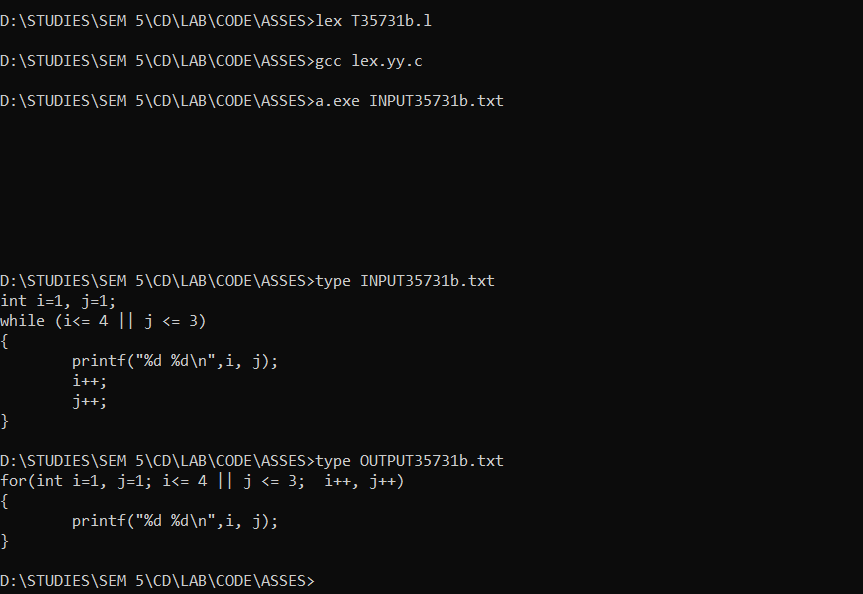
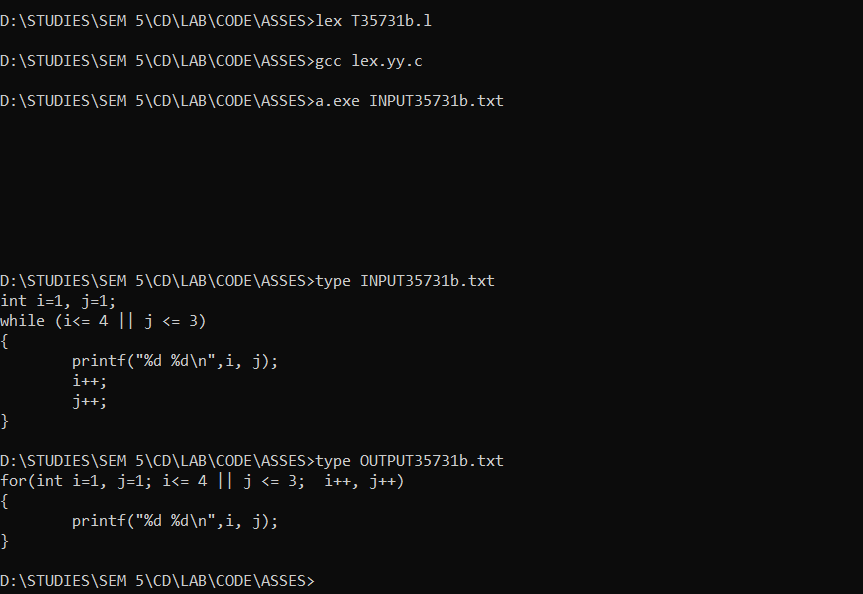
j++;

}

**LEX**



**OUTPUT**



**2. Consider the following program fragment**

inti, j, a[2][3] ;

float c , x;

for ( i = 1; i<= 10 ; i++){

for ( j = 1 ; j <=10 ; j ++ ){

a[i][j] = 1;

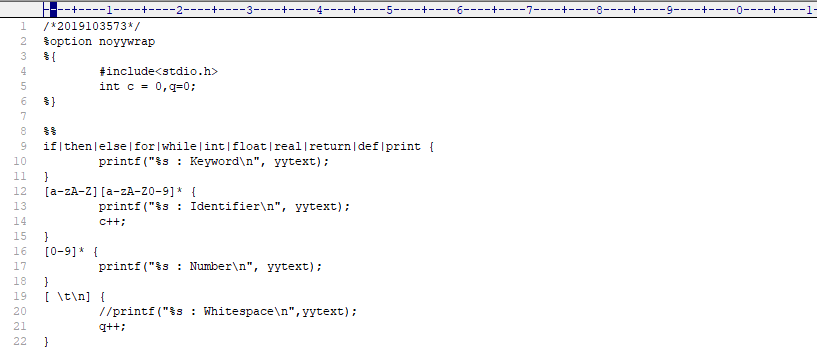
x = c + a[i][j] ;

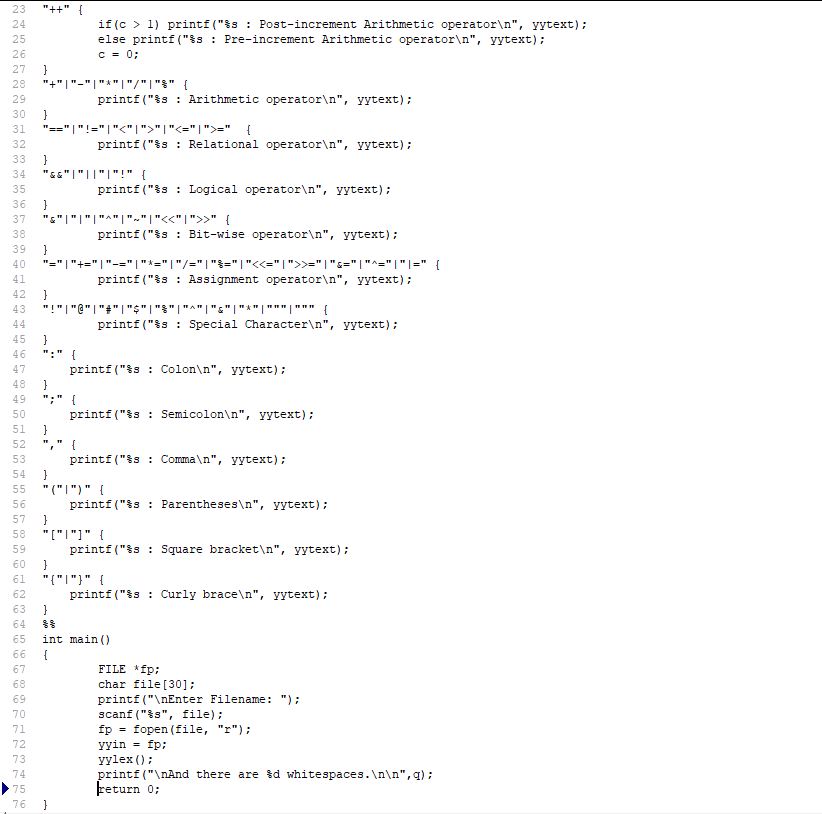
}

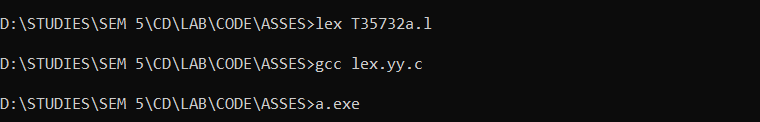
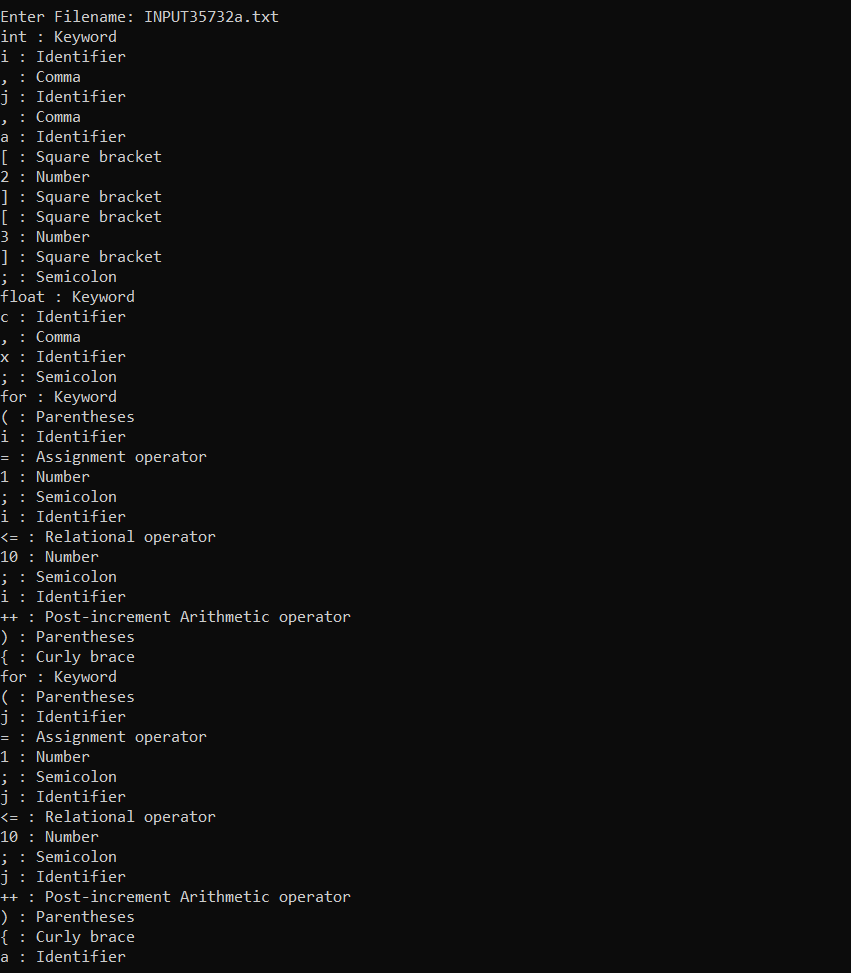
}

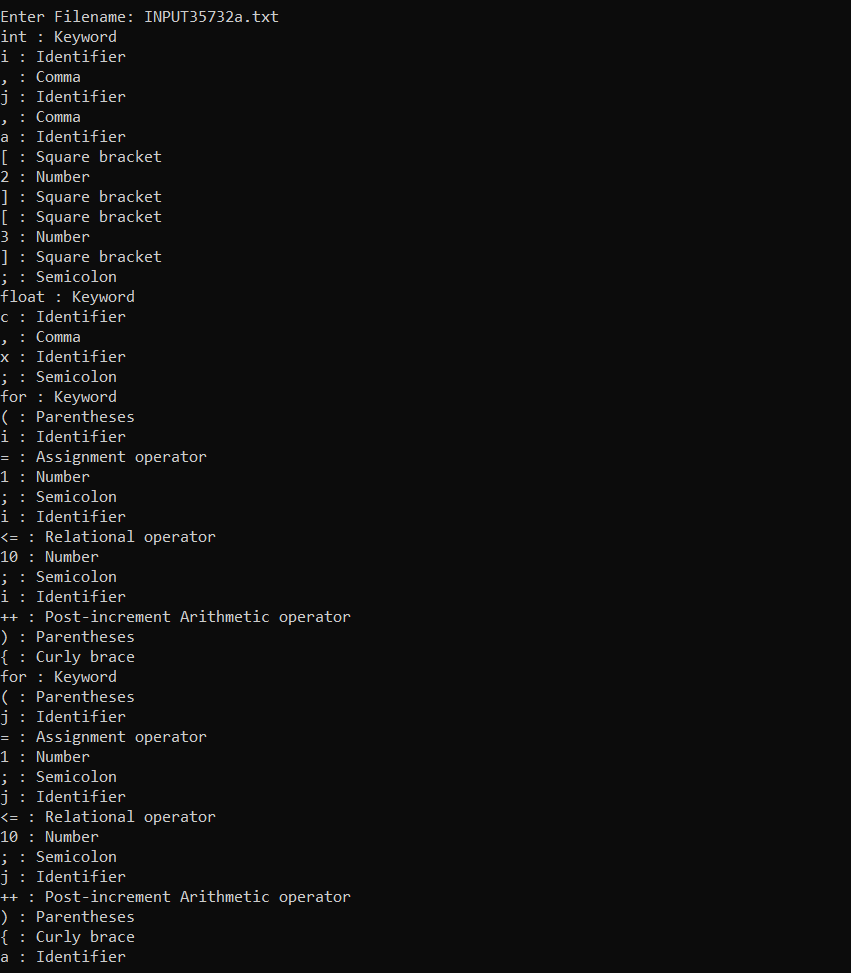
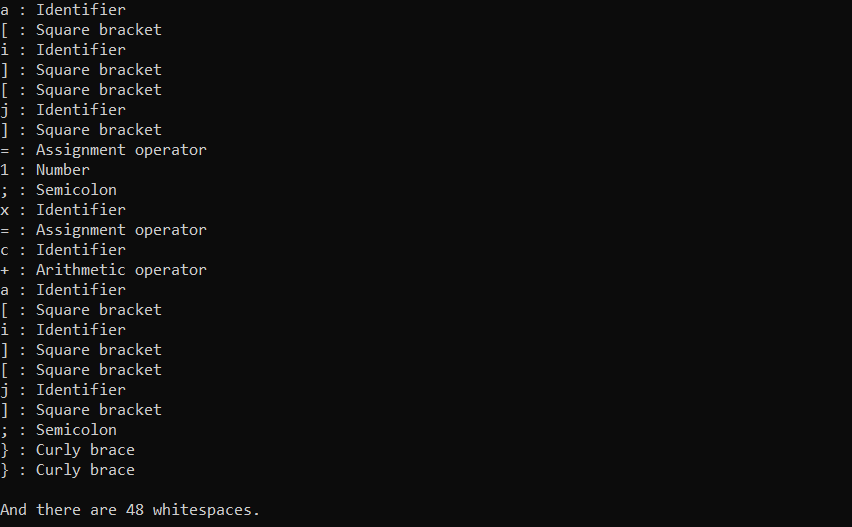
Perform the following using LEX/YACC

**a. Identify the tokens and print them**

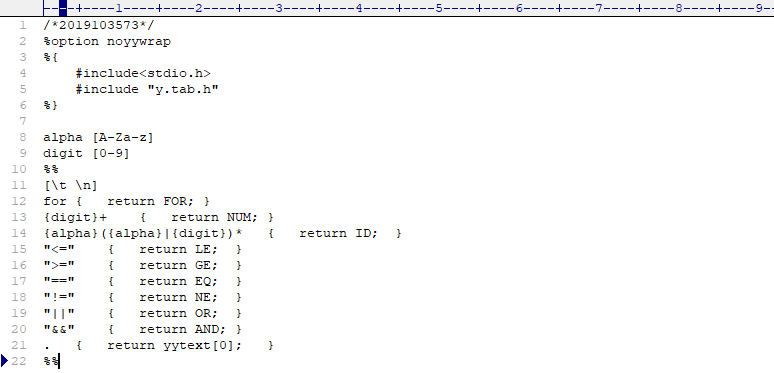
**LEX**

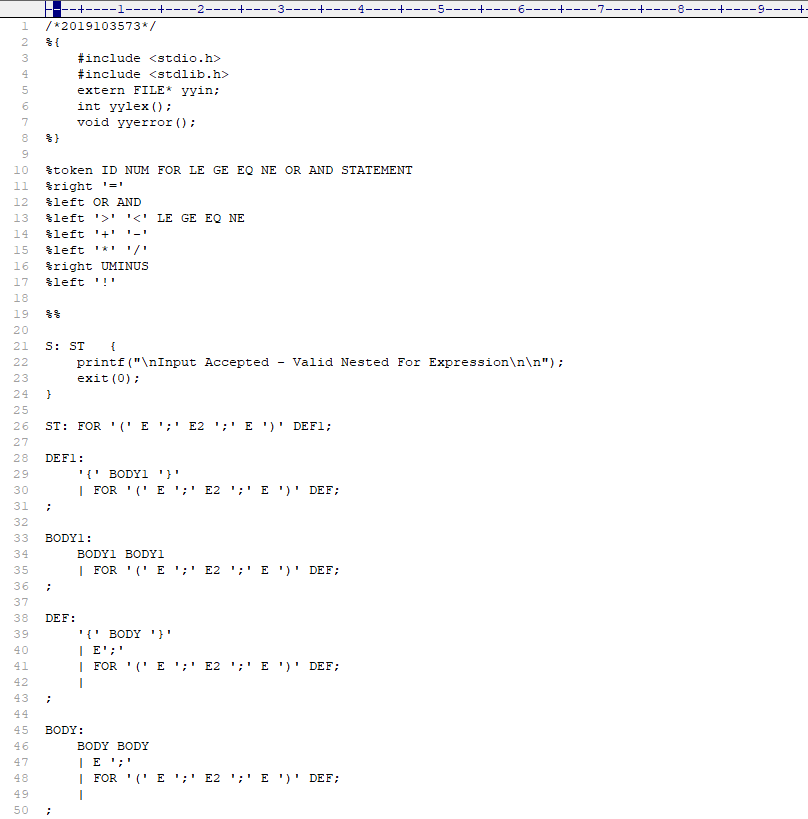


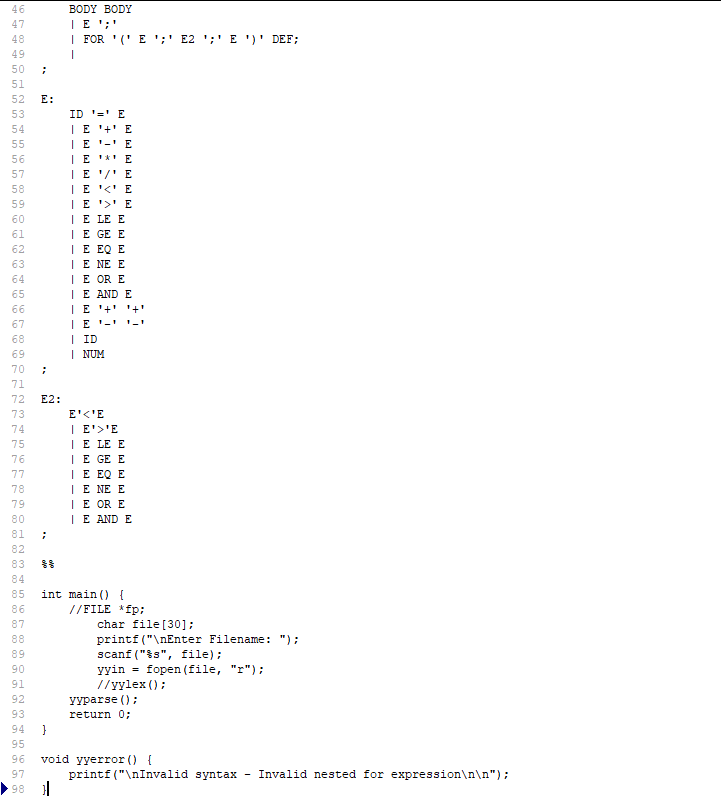
**OUTPUT**

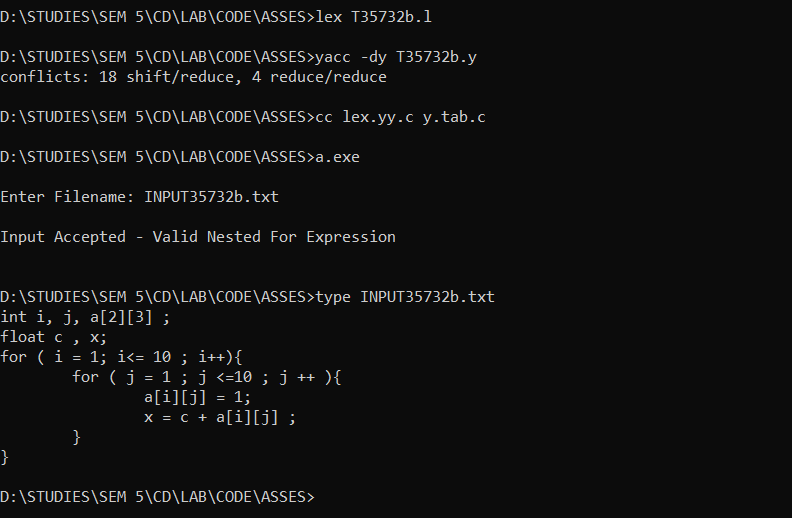
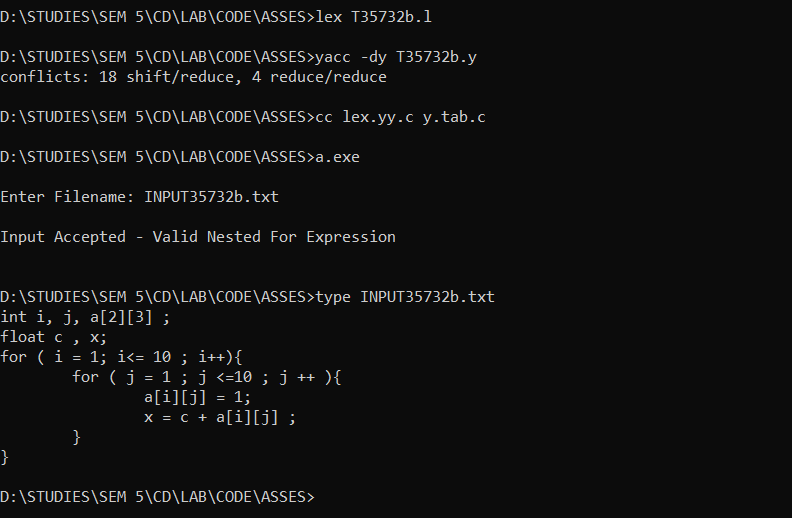


**b. Validate the constructs in the program**

**LEX**

**YACC**



**OUTPUT**

**3. Write a LEX program, which scans and stores string literals used in C language. Your lexer should detect the strings and store them into a simple symbol table and print the strings that have atleast 3 words.**

**LEX**

**OUTPUT**