Compiler Design Lab

Lab Assignment-3

Name: Dushyanth

Roll No: 18CS01009

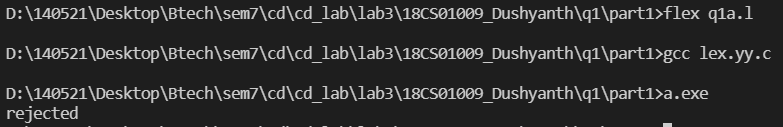
Section-1

Q 1.1 Write a LEX/Flex program that recognizes binary strings containing odd number of 0’s

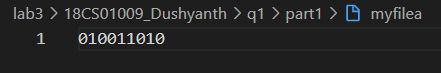
Input:



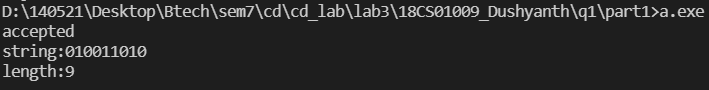
Output:



Input:



Output:

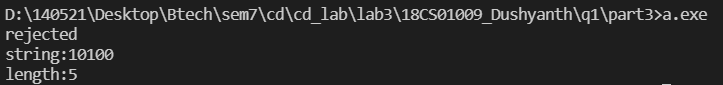


Q 1.3 Write a LEX/Flex program that recognizes binary strings whose integer equivalent is divisible by 3.

Input: (20=10100)



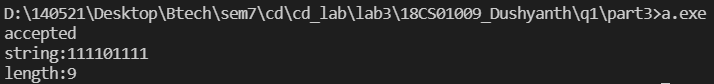
Output:



Input: (495=111101111; 495/3 = 165)



Output:



Section-2

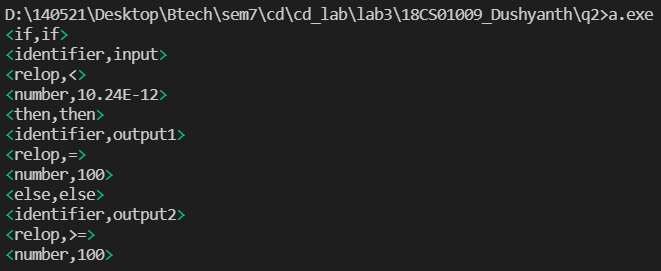
Q 2.1. Write a Lex/Flex program to describe the tokens of the above grammar, and generate a lexical analyzer using the Lex/Flex tool.

Q 2.2. Test the lexical analyzer with some input strings (You should show and explain the output of the lexical analyzer for the considered examples).

Input:



Output:



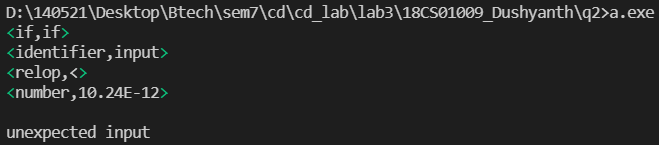
The given input: ‘if input<10.24E-12 then output1=100 else output2>=100’

Lexical analyser considers the longest prefix matched (and if there are two such matches with equal lengths then the one corresponding to the first mentioned rule is considered).

Appropriate tokens were identified based on specified pattern and the corresponding action prints **<token\_type,lexeme>**

Input: 

Output:

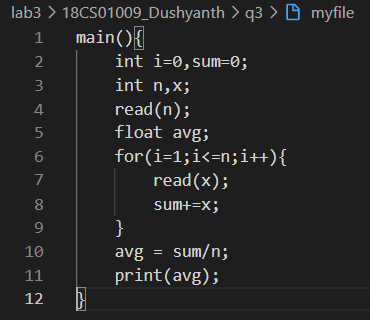


Explanation: First, the token types are identified and they are found to be: if, else, then, identifier, relop, number. After, patterns are defined to match the token types and an action is associated with the pattern.

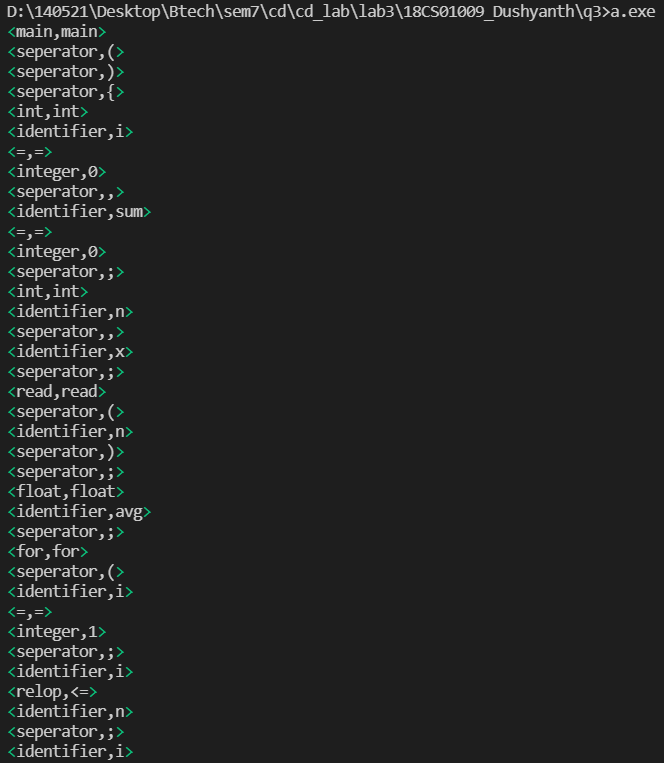
Section-3

Q 3. Construct a lexical analyzer for the following simple “C” like language using the Lex/Flex tool.

Input:



Output:

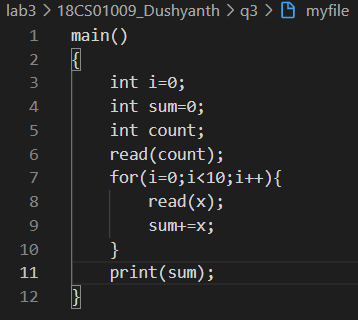




Token types considered are:

Int, float, separator, for, if, else, ++, --, arith\_op, op&assign, identifier, integer, floating\_point, =, relop, while, print, read

Input:



Output:

