**Question:** Given the grammar below, design a compiler up to the ‘Syntax Analyzer’ stage by answering the following questions:

|  |  |
| --- | --- |
| Grammar:  E → E + T | T  T → T \* F | F  T → ( E ) | digit | Input:  5 + 8 \* 6 + ( 9 + 3 ) \* 4  Output:  101 |
| 1. Write the Lex program for the given grammar | |
| 1. Write the possible output (Tokenized Format) of your written program for the input given above. | |
| 1. Identify the FIRST and FOLLOW of all the non-terminals. | |
| 1. Analyzing the grammar and the FIRST & FOLLOW sets, design the Parsing Table. | |
| 1. Using the Parsing Table, draw the Parser Tree for the given input. | |
| 1. Show the Non-recursive Parsing Table (matched, stack, input, action) for the given grammar and input. | |
| 1. Write the code of Parser Generator for the given grammar. | |
| 1. Show the calculation of output for the given input using the parsing tree   [ Hints: show how the value is passed from one step to another and also write which action is executed in each phase of the tree] | |