**Lingaya’s Institue of Management and Technology**

**B.Tech (Computer Science)**

**Compiler Design (CSE 411E)**

**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Topic** | **Date** | **Signature** |
| 1 | LEX, A Lexical Analyzer. |  |  |
| 2 | YACC, A Parser Generator |  |  |
| 3 | WAP to show the implementation of stack and it’s operations, ie. Push, Pop, and Display using Arrays. |  |  |
| 4 | WAP to check whether a given expression has equal number of (), {} and []. Also, check whether they are paired correctly and show the corresponding output. |  |  |
| 5 | WAP to count the number of operators, identifiers, (), {}, and [] in an expression and display the same. |  |  |
| 6 | WAP to read a C Program File and separate the tokens in the C Program. |  |  |
| 7 | WAP to check whether a given expression follows the grammar a^n. |  |  |
| 8 | WAP to check whether a given expression follows the grammar (a^n.b^n). |  |  |
| 9 | WAP to find the terminals and non-terminals of a given grammar. |  |  |
| 10 | WAP to find the leftmost terminal of a given grammar. |  |  |
| 11 | WAP to find the rightmost terminal of a given grammar. |  |  |
| 12 | WAP to remove left recursion from a given grammar. |  |  |
| 13 | WAP to perform left factoring on a given grammar. |  |  |
| 14 | WAP to check whether a string is generated from a given grammar or not. |  |  |