**PRACTICAL 1**

**AIM: Write a C program to identify whether a given line is a comment or not.**

**Code:**

**Output:**

**PRACTICAL 2**

**AIM: Write a C program to test whether a given identifier is valid or not.**

**Code:**

**Output:**

**PRACTICAL 3**

**AIM: Write a C program to simulate lexical analyzer for validating operators.**

**Code:**

**Output:**

**PRACTICAL 4**

**AIM: Write a C program to count the number of characters, words, spaces and lines in a given input file.**

**Code:**

**Output:**

**PRACTICAL 5**

**AIM: Write a C program to eliminate the comment lines from the program.**

**Code:**

**Output:**

**PRACTICAL 6**

**AIM: Write a C program to recognize a valid arithmetic expression and to recognize the identifiers and operators present. Print them separately.**

**Code:**

**Output:**

**PRACTICAL 7**

**AIM: Write a C program to recognize and count the number of identifiers in a given input file.**

**Code:**

**Output:**

**PRACTICAL 8**

**AIM: Write a program to left factor the given grammar.**

**Code:**

**Output:**

**PRACTICAL 9**

**AIM: Write a C program to print all numbers from the given file.**

**Code:**

**Output:**

**PRACTICAL 10**

**AIM: Write a program to remove left recursion from the given grammar.**

**Code:**

**Output:**

**PRACTICAL 11**

**AIM: Write a program which generates the quadruple table for postfix string.**

**Code:**

**Output:**

**PRACTICAL 12**

**AIM: rite a program to identify all the tokens from the source code.**

**Code:**

**Output:**