Date: 25/09/2025

Experiment 3.4

AIM

To implement recursive descent parsing

ALGORITHM

1. Start
2. Define recursiveDescent function that takes as input a grammar G pointers to strings containing the input string and expanded string and does the following:
    1. If pointer to expanded string is at the end of the string, return whether the pointer to input string is at the end of the string.
    2. If both pointers point to same character, increment both pointers and call and return recursiceDescent recursively.
    3. Let c be the current pointer.
    4. For each production rule in G of form $X \rightarrow Y_1 Y_2 .. Y_k$ where X is the pointed symbol in the expanded string:
        1. Copy the string expanded to another string.
        2. Truncate expanded at the current pointer.
        3. Append $Y_1 Y_2 .. Y_k$ to expanded.
        4. Append the suffix of the pointer from the copy to expanded.
        5. Call recursiveDescent function:
            1. If it returns true, return true.
        6. Restore expanded from the copy.
    5. Return false.
3. Read grammar.
4. Read input.
5. Initialize string expanded with only the start symbol of the grammar.
6. Call recursiveDescent for the grammar with string pointers to beginning of input and expanded.
7. Print LMD generated if string is accepted.
8. Stop

RESULT

Successfully implemented recursive descent parsing.