**Problem Statement:** Write a C program for implementing the functionalities of predictive parser for the mini language.

**G: { E-> TE′, E′ -> +TE′ | 0, T-> FT′, T′-> *FT′|0, F-> (E) | id}**

**AIM:** To write a 'C' Program to implement for the Predictive Parser (Non Recursive Descent parser) for the given grammar

**ALGORITHM / PROCEDURE:**

Input : string w$, Predictive Parsing table M
Output: A Left Most Derivation of the input string if it is valid, error otherwise.

Step1: Start

Step2: Declare a character array w [10] and Z as an array

Step3: Enter the string with $ at the end

Step4: if (A(w[z]) then increment z and check for (B(w[z])) and if satisfies increment z and check for 'd' if d is present then increment and check for (D(w[z]))

Step5: if step 4 is satisfied then the string is accepted

Else string is not

Step 6: Exit

**[Viva Questions]**

1. What is a parser and state the Role of it?
2. Types of parsers? Examples to each
3. What are the Tools available for implementation?
4. How do you calculate FIRST(),FOLLOW() sets used in Parsing Table construction?

**Exercise: 1**. Implementation of predictive parser for an Expression that generates arithmetic expressions with digits, +,*