Problem 1 – Parse trees, derivations

Consider the following grammar with start symbol A:

1. Give parse trees for the sentences (a, a) and ((a, a), a).

2. Construct a leftmost and a rightmost derivation for the sentence ((a, a), a).

Problem 2 - LL(1) and Recursive Descent Parsing

Assume the following grammar of a simple, prefix expression language.

Program ::= Stmtlist .
Stmtlist ::= Stmt NextStmt
NextStmt := ; Stmtlist | epsilon
Stmt ::= Assign | Print
Assign ::= ID = Expr
Print ::= ! ID
Expr ::= + Expr Expr | - Expr Expr | * Expr Expr | ID | ICONST
ID ::= a | b | c
ICONST ::= 1 | 2 | 3

The goal of this homework problem is to build an LL(1) parser for this simple language. Programs can be written in a single line (no new line
The homework problem consists of the following subproblems:

1. Compute the \textit{FIRST} and \textit{FOLLOW} sets for the grammar.

2. Compute the LL(1) parse table for the resulting grammar. Is the grammar LL(1) or not? Justify your answer.

3. If the resulting grammar is LL(1), show the behavior of the LL(1) skeleton parser as a sequence of states \([\text{stack content, remaining input, next action to be taken}]\) on sentence \(c=3;!c..\)

4. If the grammar is LL(1), show a recursive descent parser in a pseudo programming language, or in a language of your choice.