# CSV Postfix Notation Evaluator

This Java program processes a CSV file, where each cell may contain a numeric value, a reference to another cell (e.g., "A1"), or a postfix expression (e.g., "3 4 +"). The program evaluates each cell's value, including resolving nested cell references and postfix expressions, and then outputs the evaluated CSV content.

The method readCSV is responsible for reading data from a CSV file and storing it in a 2D list (List<List<String>>). Each list represents a row, and each element in a row list represents a cell within that row.

The main method for evaluating cells is getCellValue, which looks up a cell by its reference (like "A1") and checks if it needs to be calculated. It resolves cell references by calling itself recursively and evaluates any postfix expressions.

The fetchValueFromCell method retrieves and evaluates the value of a cell given its reference (like "A1" or "B2"). It's designed to handle different types of cell content: numeric values, postfix expressions, and cell references.

Helper functions were created to check and process elements in each cell's array value, determining if they are numbers, cell references, postfix expressions, or operators, and to perform operations when needed.

The said functions above were called in evaluateExpression method to process a postfix expression from a CSV cell. For postfix expressions, the program uses a stack: it pushes numbers onto the stack, then when it encounters an operator, it pops numbers from the stack, applies the operation, and pushes the result back.

**Limitations:**
Deeply nested postfix expressions or complex references may require additional parsing logic. The current implementation focuses on fulfilling the given technical requirements.

**Trade-offs and Design Decisions**:

The program resolves dependencies using recursion, which can be slow for deeply nested or cyclic references.