

# Algorithm for Algebraic Puzzle Solver

## Input:

- A set of algebraic equations with multiple variables (e.g.,  $x + y = 5$ ,  $2x - y = 0$ ).

## Output:

- Solutions for the variables in the equations (e.g.,  $x = 5/3$ ,  $y = 10/3$ ).
- 

## Steps:

### 1. Start:

- Begin the program.

### 2. Display Welcome Message:

- Print a welcome message and instructions for the user.
- Example:

Copy

Welcome to the Algebraic Puzzle Solver!

Enter your equations in the form ' $x + y = 5$ ' or ' $2x - y/2 = 3$ '.

Type 'done' when you're finished entering equations.

### 3. Initialize Data Structures:

- Create an empty set variables to store all variables in the equations.
- Create an empty list equations to store the symbolic equations.

### 4. Input Equations:

- While the user has not entered 'done':
  - Prompt the user to enter an equation.
  - If the input is 'done', exit the loop.
  - Otherwise:
    - Split the equation into the left-hand side (LHS) and right-hand side (RHS) using the = sign.
    - Extract all variables from the equation using symbolic parsing.
    - Add the variables to the variables set.

- Create a symbolic equation using `sympy.Eq(LHS, RHS)`.
- Add the equation to the equations list.

**5. Check for Equations:**

- If no equations were provided:
  - Print "No equations provided. Exiting."
  - **Stop.**

**6. Solve the System of Equations:**

- Use `sympy.solve(equations, variables)` to solve the system of equations.
- If a solution exists:
  - Print the solutions for each variable.
- If no solution exists:
  - Print "No solution exists."

**7. Handle Errors:**

- If any error occurs during equation parsing or solving:
  - Print an appropriate error message (e.g., "Invalid equation" or "Error solving equations").

**8. Stop:**

- End the program.
-

