

Programming Problem: Matchstick Equation Solver

In a classic matchstick puzzle, digits and operators are formed using matchsticks. By moving exactly one matchstick, an invalid equation can sometimes be transformed into a valid one.

Problem Description

Write a program that fixes an invalid arithmetic equation by moving **exactly one matchstick**.

Your program should implement a function:

```
fix_equation(equation: string) -> string
```

The input `equation` is a string representing an **invalid equation**, such as:

"6+4=4"

The equation consists of:

- Single-digit numbers (0–9)
- Operators: + or -
- Exactly one = sign

All digits and operators are assumed to be formed using standard **7-segment matchstick representations**.

Rules

1. You may **move exactly one matchstick**:
 - Remove a matchstick from one digit or operator
 - Add it to another digit or operator (possibly the same one)
2. After the move:
 - All symbols must still form **valid digits or operators**
 - The resulting equation must be **mathematically correct**
3. You may transform:
 - One digit into another (e.g., $6 \rightarrow 8$, $9 \rightarrow 3$)
 - Operators if applicable (e.g., $+$ $\leftrightarrow$ $-$ if representable)
4. Output:
 - Return a **valid equation string** after one move
 - If multiple solutions exist, return any one of them
 - If no solution exists, return "No solution"

Examples

Input: "6+4=4"

Output: "8-4=4"

Input: "9-3=3"

Output: "8-3=5"

Requirements

- Explicitly model the matchstick structure of digits (e.g., using 7-segment encoding)
- Systematically explore all valid one-match transformations
- Clearly document your approach

Bonus (Optional)

- Allow multiple matchstick moves
- Support multi-digit numbers (e.g., " $12+3=9$ ")
- Return all possible solutions instead of just one