```c
#include <stdio.h>
#include <string.h>

struct Var {
    char name;
    int value;
    int isConstant;
} symbolTable[10];

int varCount = 0;

int getConstantValue(char varName, int *foundPtr) {
    for (int i = 0; i < varCount; i++) {
        if (symbolTable[i].name == varName && symbolTable[i].isConstant) {
            *foundPtr = 1;
            return symbolTable[i].value;
        }
    }
    *foundPtr = 0;
    return 0;
}

void updateVariable(char varName, int val, int isConst) {
    for (int i = 0; i < varCount; i++) {
        if (symbolTable[i].name == varName) {
            symbolTable[i].value = val;
            symbolTable[i].isConstant = isConst;
            return;
        }
    }
    symbolTable[varCount].name = varName;
    symbolTable[varCount].value = val;
    symbolTable[varCount].isConstant = isConst;
    varCount++;
}

int calculateResult(char op, int leftOperand, int rightOperand) {
    switch (op) {
        case '+': return leftOperand + rightOperand;
        case '-': return leftOperand - rightOperand;
        case '*': return leftOperand * rightOperand;
        case '/':
            if (rightOperand != 0) return leftOperand / rightOperand;
            return 0;
        default: return 0;
    }
}

int main() {
    char inputLine[50];
    char targetVar, op;
    char leftVar, rightVar;
    int immediateVal;

    printf("Enter assignments (ex: a = 5, b = a + 3). Type 'end' to stop.\n\n");
    while (1) {
        if (fgets(inputLine, sizeof(inputLine), stdin) == NULL) break;
        if (strncmp(inputLine, "end", 3) == 0) break;
        int scanCount;

        // 1. Case: targetVar = immediateVal (e.g., a = 5)
        scanCount = sscanf(inputLine, " %c = %d", &targetVar, &immediateVal);
        if (scanCount == 2) {
            updateVariable(targetVar, immediateVal, 1);
            continue;
        }

        // 2. Case: targetVar = leftVar op immediateVal (e.g., b = a + 3)
        scanCount = sscanf(inputLine, " %c = %c %c %d", &targetVar, &leftVar, &op, &immediateVal);
        if (scanCount == 4) {
            int isLeftConst;
            int leftVal = getConstantValue(leftVar, &isLeftConst);

            if (isLeftConst) {
                int result = calculateResult(op, leftVal, immediateVal);
                updateVariable(targetVar, result, 1);
            } else {
                updateVariable(targetVar, 0, 0);
            }
            continue;
        }

        // 3. Case: targetVar = leftVar op rightVar (e.g., c = a * b)
        scanCount = sscanf(inputLine, " %c = %c %c %c", &targetVar, &leftVar, &op, &rightVar);
        if (scanCount == 4) {
            int isLeftConst, isRightConst;
            int leftVal = getConstantValue(leftVar, &isLeftConst);
            int rightVal = getConstantValue(rightVar, &isRightConst);

            if (isLeftConst && isRightConst) {
                int result = calculateResult(op, leftVal, rightVal);
                updateVariable(targetVar, result, 1);
            } else {
                updateVariable(targetVar, 0, 0);
            }
        }
    }

    printf("\n--- Final Variable Status (Constant Propagation) ---\n");
    for (int i = 0; i < varCount; i++) {
        if (symbolTable[i].isConstant)
            printf("%c = %d\n", symbolTable[i].name, symbolTable[i].value);
        else
            printf("%c = (non-constant)\n", symbolTable[i].name);
    }

    return 0;
}

// OUTPUT
Enter assignments (ex: a = 5, b = a + 3). Type 'end' to stop.

a=5
b=a+7
c=b*4
d=c/b
e=d-5
end

--- Final Variable Status (Constant Propagation) ---
a = 5
b = 12
c = 48
d = 4
e = -1
```