import java.util.\*;

public class CSP {

public static void main(String[] args) {

String[] words = {"WAIT", "ALL", "GIFTS"};

List<Character> uniqueLetters = new ArrayList<>();

for (String word : words) {

for (char letter : word.toCharArray()) {

if (!uniqueLetters.contains(letter)) {

uniqueLetters.add(letter);

}

}

}

int[] digits = new int[uniqueLetters.size()];

boolean[] used = new boolean[10];

boolean solutionFound = solveCryptarithmetic(words, uniqueLetters, digits, used, 0);

if (solutionFound) {

System.out.println("Solution found:");

for (int i = 0; i < uniqueLetters.size(); i++) {

System.out.println(uniqueLetters.get(i) + " = " + digits[i]);

}

} else {

System.out.println("No solution found.");

}

}

static boolean solveCryptarithmetic(String[] words, List<Character> uniqueLetters, int[] digits, boolean[] used, int index) {

if (index == uniqueLetters.size()) {

return evaluateExpression(words, uniqueLetters, digits);

}

for (int i = 9; i >= 0; i--) {

if (!used[i]) {

used[i] = true;

digits[index]=i;

if (solveCryptarithmetic(words, uniqueLetters, digits, used, index + 1)) {

return true;

}

used[i] = false;

}

}

return false;

}

static boolean evaluateExpression(String[] words, List<Character> uniqueLetters, int[] digits) {

int[] wordValues = new int[words.length];

for (int i = 0; i < words.length; i++) {

int value = 0;

for (char letter : words[i].toCharArray()) {

int index = uniqueLetters.indexOf(letter);

value = value \* 10 + digits[index];

}

wordValues[i] = value;

}

int sum = 0;

for (int i = 0; i < wordValues.length - 1; i++) {

sum += wordValues[i];

}

return (sum == wordValues[wordValues.length - 1]);

}

}