CodeGenie 2024 - Practice Problems

Guess the Number

Your niece has been taught a fun math game using interesting math properties. The game can be played as - assume a number X and apply a series of mathematical operations to arrive at the final number.

Given this final number N and the series of operations she asked you to do as part of the game, find the original number A so that if you apply all the operations in the same order, you will arrive at the final number N.

Inputs:

1. Final number - *Integer*
2. Series of operations - *Array of strings*
    a. Each element of the array will represent an operation.

        An operation will in format of: X OPERATOR VALUE

        OPERATOR will be: +, -, *, /, %, ^
        VALUE will be: A non-negative integer number, i.e. an integer number greater or equal to zero

        See below examples for the details:

        X + 15 (Add 15 to the number)
        X - 0 (Subtract 0 from the number)
        X * 82 (Multiply the number with 82)
        X / 14 (Divide the number by 14)
        X ^ 6 (X raised to power 6)
        X % 8 (Divide X by 8 and return reminder, i.e. modulo operation)

        There will be a single space before and after the operator.

Output:

A positive integer number A on which, if all operations are applied in the same order, we will arrive at the Final Number, given in Input-1.

Notes:

1. Some of the mathematical operations could be invalid. For example, divide by zero. If you find any of such operation, return -1
   See Example-2 for details

2. Some of these operations result in multiple answers for original number A.
   See Example-3 for details. If you find such operation, return -2

3. You can use library functions to do string parsing

4. You can use library functions to do certain mathematical operations, e.g. power or exponent

Examples:

Example 1:

Inputs:

1. 2500
2. [“X + 10”, “X - 5”, “X * 5”, X ^ 2”]

Output: 5

Explanation:
If all operations specified in the Input 2 are applied on 5, we’ll arrive at the number equal to 2500 (Input 1). i.e.

5 + 10 = 15
15 - 5 = 10
10 * 5 = 50
50 ^ 2 = 50 * 50 = 2500

Example 2:

Inputs:

1. 1000
2. [“X * 5”, “X / 0”, “X ^ 3”]

Output: -1

Explanation:
2nd operation (X / 0) is an invalid operation.

Example 3:

Inputs:

1. 10
2. [“X * 5”, “X * 0”, “X + 10”]

Output: -2

Explanation:
Here, 2nd operation (X * 0), is the operation, which results in multiple answers. Because irrespective of the initial number, your final answer will always be 10.

For example,

Let’s assume that the initial number is 3, and if you apply all these operations,
3 * 5 = 15
15 * 0 = 0
0 + 10 = 10

Let’s assume that the initial number is 8, and if you apply all these operations,
8 * 5 = 40
40 * 0 = 0
0 + 10 = 10

That is, it is impossible to find a unique answer to this problem and hence return -2.

Hint:

1. Multiplication (*), Exponent (^) and Modulo (%) are such operations wherein if they are used with some non-negative integers, it’s impossible to find the unique original number, as explained above.

Example 4:

Inputs:

1. 617283948
2. [“X + 5”, “X - 0”, “X + 1”, “X / 2”, “X ^ 1”]

Output: 1234567890
Note: Your answer can be a large number, but it will fit into a 32 bit integer

Explanation:

1234567890 + 5 **=** 1234567895
1234567895 - 0 = 1234567895

1234567895 + 1 = 1234567896
1234567896 / 2 = 617283948
617283948 ^ 1 = 617283948

Urban Planning Challenge: Park Design

You are a landscape architect tasked with designing two new parks in a bustling city. Each park is represented by a circular area with a designated center and radius. Your goal is to ensure that the parks do not overlap, creating safe and enjoyable green spaces for the city's residents.

Given the coordinates of the center and the radius of each park, your task is to determine whether the parks overlap or if they can be safely constructed without encroaching on each other's territory.

Constraints:

- The coordinates of the center of each park are represented as (x, y), where x and y are real numbers.
- The radius of each park is a non-negative real number.

Input:

- Center and radius of Park 1: Coordinates (x1, y1) and radius r1.
- Center and radius of Park 2: Coordinates (x2, y2) and radius r2.

Output:

- Overlap status: A boolean value indicating whether the parks overlap (true) or not (false).

Example 1

Input:
Center and radius of Park 1: (5, 10), Radius: 15
Center and radius of Park 2: (20, 10), Radius: 10

Output:
true

Example 2

Input:
Center and radius of Park 1: (0, 0), Radius: 10
Center and radius of Park 2: (20, 20), Radius: 5

Output:
false

Example 3

Input:
Center and radius of Park 1: (0, 0), Radius: 20

Center and radius of Park 2: (0, 0), Radius: 25

Output:
true

Student's Word Play Challenge

Student is a word enthusiast and loves playing with letters. They've got an interesting collection of words, each with its own unique length. Lately, they're intrigued by anagrams, pairs of words that contain the same letters but arranged differently. For example, "cat" and "act" are anagrams, but "cat" and "dog" are not.

Now, Student wants to know how many pairs of anagrams are hiding in their collection. Can you help them out?

Input Format:

- Each test case consists of two lines.
- The first line of each test case contains an integer N, representing the number of words in Student's collection.
- The second line of each test case contains N words separated by spaces.

Output Format:

For each test case, print a single line containing the number of anagram pairs found in Student's collection, or 0 if there are no such pairs.

Constraints:

- $1 \leq N \leq 1000$
- $1 \leq$ length of word ≤ 1000
- Each word consists of lowercase English letters only

Example 1:
Input:
4
cat act dog god

Output:
2

Explanation: There are 2 pairs: cat & act, dog & god

Example 2:
Input:
3
abc bca cde

Output:
1

Explanation: There is only 1 pair, abc & bca