BROCODE REPORT:

==

```
1. sup {
   var a = int(listen("")); //
   var b = int(listen("")); //

   // Case #1: 10 + 20
   fr (a is_vibe 10 && b is_vibe 20) { //
      spit(30); //
   }
   // Case #2: 0 + 0
   meh (a is_vibe 0 && b is_vibe 0) { //
      spit(0); //
   }
   // Case #3: -5 + 10
   meh (a is_vibe -5 && b is_vibe 10) { //
      spit(5); //
   }
   // Case #4: 999 + 1
   meh (a is_vibe 999 && b is_vibe 1) { //
      spit(1000); //
   }
   // Case #5: -100 + -50
   meh (a is_vibe -100 && b is_vibe -50) { //
      spit(-150); //
   }
   // General fallback logic


   peace; //
}
```

PROBLEM: What if someone hardcodes the program based on the missed test cases they see after first wrong test run?

===

ONE MORE:

I clicked on run test for an easy question, and then switched to a hard question.
The hard question was marked solved but only +2 was given.
But that easy question, is now unsolvable, as in, there is no tick

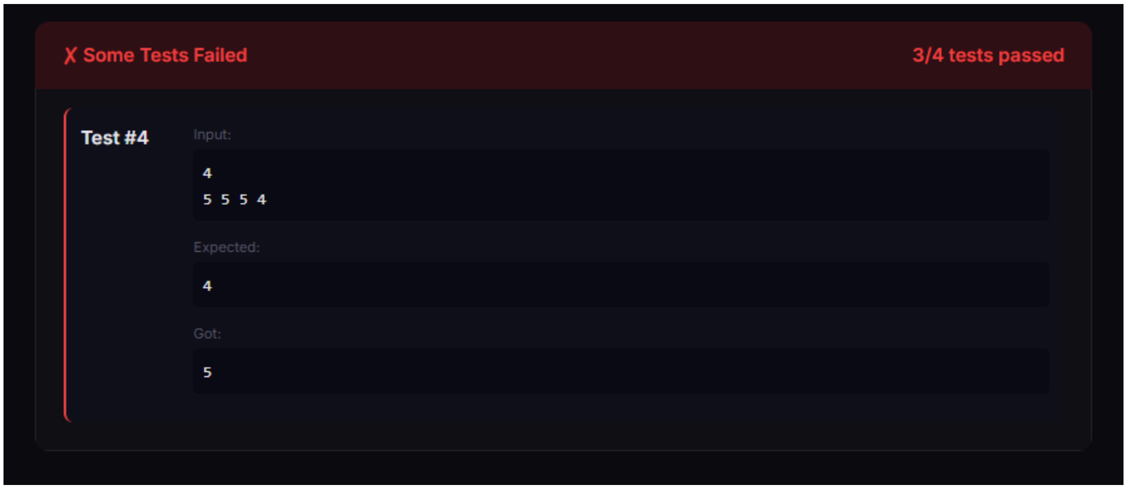
============

H08:

Gemini gave this code:

```
sup {
    var n = int(listen("")); // Read the number of elements
    var input_str = listen(""); // Read the line of N integers

    // Convert input string into a Squad of integers
    var string_squad = split(input_str, " "); // Split by space
    var numbers = []; // Create a new squad to store integers

    cycle (var s in string_squad) { // Iterate through the strings
        numbers.push(int(s)); // Convert to int and add to the squad
    }

    // Use the built-in sort function (returns a new sorted squad)
    var sorted_numbers = sort(numbers); // Sorts in ascending order

    // The second largest is at index (length - 2)
    var size = len(sorted_numbers); // Get total size
    var second_largest = sorted_numbers[size - 2]; // Access second-to-last element

    spit(second_largest); // Output the result

    peace; // Exit the program
}
```

OUTPUT:



✗ Some Tests Failed

3/4 tests passed

Test #4

Input:

```
4
5 5 5 4
```

Expected:

```
4
```

Got:

```
5
```

==================================

H04

```
sup {
    // Read both strings
    var str1 = listen(""); //
    var str2 = listen(""); //

    // Logic: If lengths differ, they can't be anagrams
    fr (len(str1) != len(str2)) {
        spit("No");
    }
    nah {
        // Convert strings to squads of characters
        var squad1 = split(str1, ""); //
        var squad2 = split(str2, ""); //

        // Sort the squads to align characters alphabetically
        var sorted1 = sort(squad1); //
        var sorted2 = sort(squad2); //

        // Check if the sorted squads are an identical vibe
        fr (sorted1 is_vibe sorted2) {
            spit("Yes");
        }
        nah {
            spit("No");
        }
    }
    peace; //
}
```

Output”

```
Runtime Error: Internal Error: empty separator
```

Gemini solution to this:

```
sup {
    var str1 = listen("");
    var str2 = listen("");

    fr (len(str1) != len(str2)) {
        spit("No");
    }
    nah {
        var squad1 = [];
        var squad2 = [];

        // Manually push each character into squads
        cycle (var char in str1) {
            squad1.push(char);
        }
        cycle (var char in str2) {
            squad2.push(char);
        }

        // Sort the manual squads
        var sorted1 = sort(squad1);
        var sorted2 = sort(squad2);

        // Final comparison
        fr (sorted1 is_vibe sorted2) {
            spit("Yes");
        }
        nah {
            spit("No");
        }
    }

    peace;
}
```

“ split(str, "") is not allowed “

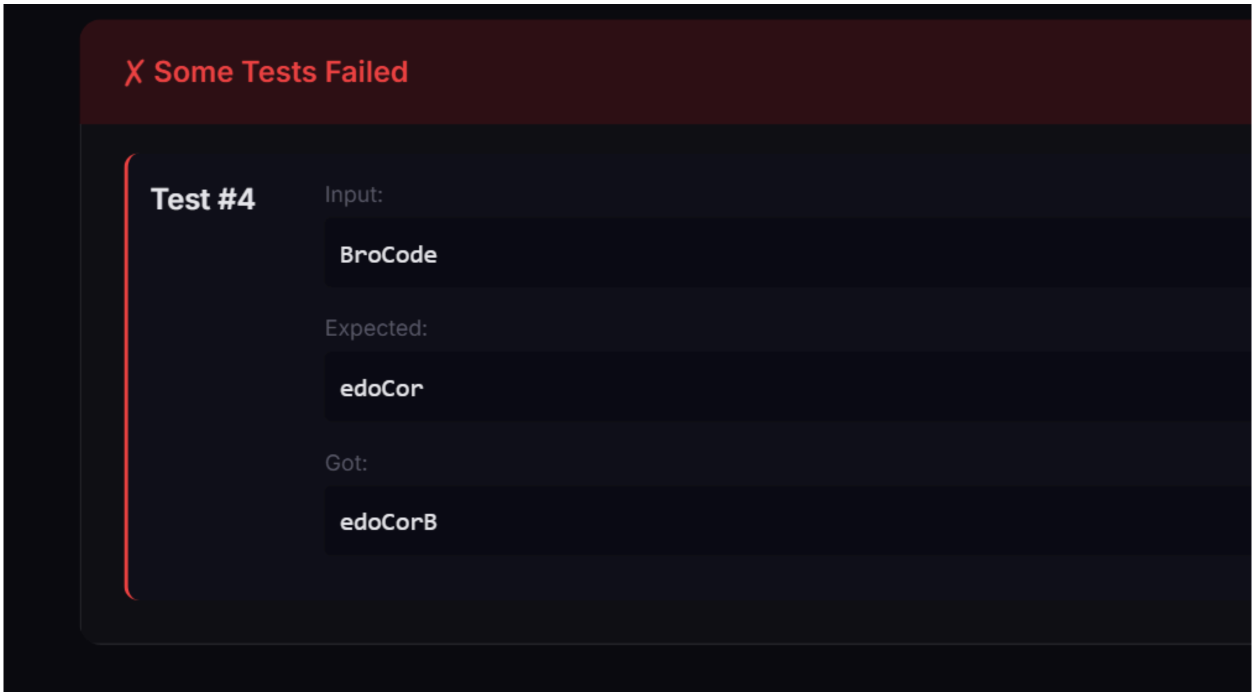
====================================
H09

“”The tests failed because of **Integer Division**. In Bro Code, using the `/` operator converts the result into a float (e.g., `2.0`), but the expected output is a strict integer (`2`). To fix this, you should use the floor division operator `//` or wrap the final calculation in `int()`.””

=========================

M01

```
sup {
    var text = listen("");
    var reversed = "";
    var length = len(text);

    // Start from the last index and go to 0
    cycle (var i in range(length - 1, -1, -1)) {
        reversed = reversed + char_at(text, i);
    }

    spit(reversed);
    peace;
}
```



✗ Some Tests Failed

Test #4

Input:

```
BroCode
```

Expected:

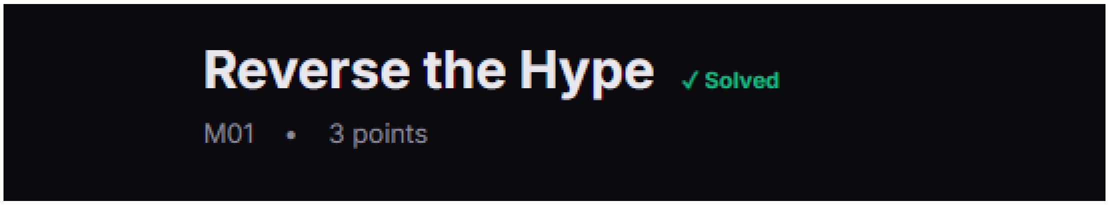
```
edoCor
```

Got:

```
edoCorB
```

GEMINI IS CORRECT, test case is wrong, this particular one.

But as said earlier



Reverse the Hype ✓ Solved

M01 • 3 points

It shows solved. Only UI problem tho ig

SOLVED: If you refresh, not an issue.

==