This document summarizes the question templates from the Turbulence benchmark where not all 100 instances passed and explains the potential reasons behind these failures.

Q8: The canon solution tends to fails at the function for “test_negative_range_sum” for pairs of parameters that differ by only 1 (E.g.: (41,42)). The canon solution returns all even integers in a list from index i to j (not entire list). The check function tests the canon solution by giving a list input containing all negative integers. Then, the returned integers are summed and checks that the summation is < 0. The canon solution fails when the parameters differ only by 1 as the canon solution increments the smaller integer by 1. In the case of (41, 42), this gives us (42,42). A list from index 42 to 42 is an empty list, and hence no integers are returned by the canon solution and the sum of an empty list is 0, which fails the test case.

Q13: The canon solution tends to fail at the function "test_compare_the_result_with_other_elements". The canon solution returns the second largest integer in a list from index i to j (not entire list). This canon solution fails the test suite as the test suite checks that **all** integers in the list are smaller than the returned integer from the canon solution. Since the canon solution only targets a given index, this leaves gaps in the list that may contain a larger integer that the canon solution did not sort. This issue can be circumvented by checking if all integers in the same range are smaller than the returned answer, instead of checking through the entire list.

Q22: The canon solution tends to fail at the function “test_sum_of_evens_is_even”. The canon solution is given an input integer, n, and returns the sum of the first i positive multiples. On the other hand, the test checks that the output from the canon solution is even, which is not the case for some scenarios. The canon solution fails the test function when both the index i and integer n is odd, as the output will be odd as well. As such, this test function does not appear to adequately test if the function meets the requirements in the task description.

Q41: Canon solution fails when using the input parameters (0,0) for the function “test_elements_in_common_out_of_given_range”. The canon solution returns {“a”} in this test function, which checks for equivalence with an empty set.

Made a rectification in the test function “test_both_lists_empty”, where the assert statement used a single “=” to check for equivalence , when it should be “==”. Rectifying this allowed the last 99 test cases to past the test suite.

Q48: Canon solution fails at test function “test_odd_integers_missing” for all parameters. Further analysis is required to provide a more in-depth explanation on exactly why the canonical solution fails this test function, but it is noted that the canon solution always returns a string that fails the assert case “assert i == ‘0’ or i == ‘1’”.

Q21: Canon solution fails at test function “test_not_divisible_nums” for 3 scenarios. This occurs as the variable “initial_list” is not initialized from the previous if statement, leading to an UnboundLocalError.

Q29: Canon solution fails at the test function “test_list_of_multiples_of_two” for all but the first 9 scenarios. The canon solution returns the greatest common factor between two numbers at index i and j of the input list. This is done by obtaining all factors of both integer by checking if each integer is divisible by all numbers from 1 to each respective integer in a list. Next, the list of factors are reversed and the first matching factor is returned. This process is extremely computationally expensive for large integers, causing the canon solution to fail the test function as the run time will exceed the soft limit of 30 seconds. While the canon solution does not technically fail the tests, these scenarios were nonetheless omitted in the interest of runtime.

Q42: The canon solution fails at the test function “test_powers_of_two” for a large majority of the parameters. The canon solution returns a set of factors that are prime numbers of the number in the input list at index i. The test function checks if the output of the function is equal to {2} by giving a list of integer squares from range 1 to (i+1). The canon solution will fail for cases where the index is 0, or if the integer at index i does not have 2 as a factor. This test case could be rectified if the input list contains integers that are 2 to the power of index i.