**Report on Fixes – Score Trend Analyzer Program**

**Introduction**

This report explains the corrections made to the original *Score Trend Analyzer* Python program. The initial version of the program contained syntax errors and logical mistakes. These errors were identified and fixed to ensure the program runs correctly and produces accurate results.

---

**Total Errors Identified in Original Code: 3**

1. Indentation Error
2. Missing Closing Parenthesis
3. Logical Error in Average Calculation

---

**Fix 1: Indentation Error**

**Problem**

In the original code, the statement inside the while loop was not properly indented. Python requires indentation to define code blocks inside loops and conditions.

**Fix**

All statements inside the while loop were properly indented using four spaces.

**Result**

The program now runs without an IndentationError.

---

**Fix 2: Missing Closing Parenthesis**

**Problem**

The line:

print("Improvement count:", improved

was missing a closing bracket ).

**Fix**

The missing parenthesis was added:

print("Improvement count:", improved)

**Result**

The SyntaxError was removed and the program executes correctly.

---

**Fix 3: Logical Error in Average Calculation**

**Problem**

The average was calculated using:

average = previous / tests

The variable previous only stored the last score, not the total of all scores. This produced an incorrect average.

**Fix**

A new variable total was introduced to store the sum of all scores:

total = total + score

average = total / tests

**Result**

The average score is now calculated correctly.

---

**Additional Improvement**

The updated version introduced:

previous = None

and used:

if previous is not None:

This prevents incorrect comparison of the first score with zero and makes the program more accurate and professional.

---

**Conclusion**

After fixing the three main errors and improving the logic, the program now:

- Runs without syntax errors
- Correctly tracks score improvement
- Calculates accurate average
- Handles the first score properly

The final version of the program is logically correct, efficient, and structured properly according to Python programming standards.