Lab 3 — Debugging & Troubleshooting Report

Student: William Hannah
Assignment: Single A – Lab 3
Objective: Identify, debug, and correct syntax and logic errors across multiple scripts using proper debugging tools and methodology.

Overview

This lab required debugging and correcting errors across multiple programming languages, including **HTML, Python, JavaScript, and Shell scripting**. The goal was not to rewrite the programs, but to **identify errors, apply targeted fixes, and document the troubleshooting process**.

Debugging was performed using **VS Code**, **browser Developer Tools**, and language-specific error messages.

Debugging Tools & Methodology

Tools Used

- **Visual Studio Code**
- **Browser Developer Tools**
- **Breakpoints**
- **Debugger statements**
- **Prettier (formatter)**

Debugging Process

1. Opened the lab files in **VS Code**.
2. Ran each script to observe errors or unexpected behavior.
3. Used **breakpoints** by clicking in the gutter (left of line numbers).
4. Inserted `debugger;` statements when necessary.

5. Stepped through execution using:
    - Step Over
    - Step Into
    - Resume
6. Read console output and error messages carefully.
7. Applied incremental fixes and re-tested after each change.

File: `index4.html` — Game Scorecard

Issues Identified & Fixes Applied

- **Line 32:** Removed an extra = causing a syntax issue.
- **Lines 42–43:** Swapped `max` and `min` values (logic error).
- **Line 82:** Corrected another `max` / `min` reversal.
- **Line 91:** Fixed incorrect usage of the `else` keyword.
- **Line 10:** Changed increment to `index--;` to correct loop logic.
- **Line 119:** Removed an extra closing `}` that broke execution.

Notes

- **Prettier** was installed to properly format HTML and JavaScript, improving readability and reducing syntax issues.
- Debugging was verified using browser Developer Tools with breakpoints.

File: `Python_app4.py`

Issues Identified & Fixes Applied

- **Line 19:** Changed `int` to `float` to support decimal values.
- **Line 35:** Replaced division by `0` with `len(...)` to prevent runtime errors.
- **Line 80:** Replaced bare `except:` with `except Exception:` to follow best practices.
- **Line 91:** Added missing `:` after a conditional statement.
- **Line 92:** Changed + to − to fix incorrect counter logic.
- **Line 97:** Added missing `:` after parentheses.

Additional Observations

- Some commented-out lines required shortening due to editor warnings.
- VS Code requested additional spacing between major blocks of code, likely due to linting or formatting rules.
- Only a small subset of the team experienced these warnings, suggesting environment or editor differences.

File: `Script4.js`

Issues Identified & Fixes Applied

- **Line 26:** Changed `Infinity` to `-Infinity` for correct initialization.
- **Line 27:** Removed − from `Infinity` where incorrect.
- **Line 55:** Added missing `}` to close a code block.
- **Line 71:** Added strict equality (===) operator.
- **Line 92:** Added `-1` after `.length` to reference the last index.
- **Line 100:** Changed increment to `cnt--` to fix loop behavior.
- **Line 110:** Corrected conditional structure to `} else {`.

File: `Script4.sh`

Issues Identified & Fixes Applied

- **Line 24:** Replaced division by `0` with `line_count`.
- **Line 39:** Added quotes around `"$w"` to prevent word splitting.
- **Line 42:** Added quotes around `"$longest"` for safety.
- **Line 43:** Added missing `}` to close function.
- **Line 65:** Changed increment to decrement (`-1`) to fix loop logic.
- **Line 75:** Added missing `;` after `]` in conditional statement.

Conclusion

Through systematic debugging using breakpoints, error messages, and incremental testing, all syntax and logic errors were successfully identified and corrected. This lab reinforced the importance of:

- Reading error messages carefully
- Using debugging tools effectively
- Applying small, targeted fixes instead of rewriting code
- Understanding language-specific best practices

All scripts now execute correctly and meet the requirements of the assignment.

Reflection

This lab helped me better understand how to systematically debug code rather than relying on trial and error. By using breakpoints, developer tools, and error messages, I was able to identify both syntax and logic issues across multiple programming languages. The experience reinforced the importance of reading errors carefully, testing changes incrementally, and following best practices specific to each language. Overall, this assignment improved my confidence in troubleshooting unfamiliar codebases and strengthened my problem-solving skills.