1. Assignment Description

This assignment employs practicing the principles of Pairwise and Pairwise Orthogonal Arrays in order to reduce redundancy in testing.

---

2. Author- Ryan Davis

---

3. Summary

Results:

I. Part 1

1. Total number of test cases:

- Operating System: 3 options (Mac iOS, Linux, Windows 11)
- Browser: 3 options (Safari, Firefox, Chrome)
- Student Type: 2 options (Undergraduate, Graduate)
- Student Location: 2 options (On Campus, Remote)

Total test cases = 3x3x2x2 = 36 test cases

2.

Test Cases	OS	Browser	Student Type	Student Location
1	iOS	Safari	U	C
2	iOS	Firefox	G	R
3	iOS	Chrome	U	R
4	Linux	Safari	G	C
5	Linux	Firefox	U	R
6	Linux	Chrome	G	C
7	Win11	Safari	U	R
8	Win11	Firefox	G	C
9	Win11	Chrome	U	C

3. A={3,3,2,2}
4. $L_9(3^3 2^2)$

Test Cases	OS	Browser	Student Type	Student Location
1	iOS	Safari	U	C
2	iOS	Firefox	G	R
3	iOS	Chrome	U	R
4	Linux	Safari	G	C
5	Linux	Firefox	U	R

6	Linux	Chrome	G	C
7	Win11	Safari	U	R
8	Win11	Firefox	G	C
9	Win11	Chrome	U	C

5. All combination testing is exhaustive and redundant. There is complete coverage, however the testing is inefficient. Pairwise testing allowed for 9 total test cases, covering all possible pairs of variable values. The pairwise-OA testing allowed for the same amount of test cases, ensuring each pair appears once in testing.

II. Part 2

1. Total test cases = 2x2x2x2 = 16 test cases
2. $L_8(2^4)$
3. $L_8(2^7)$
4. An L8 pairwise orthogonal array covers 8 test cases.

III. Part 3

1. Total test cases = 3x4x3 = 36 test cases
2. The minimum number of tests for pairwise testing is 4, accounting for the 4 classes of books (largest number of values for a variable).
3. $L_9(3^34^1)$
4. The mixed alphabet is helpful for accounting for different levels in variables. It does add additional burden as an orthogonal array for mixed alphabet becomes complex. The mixed alphabet is advantageous test cases are reduced significantly. Also testing becomes balanced, as test cases are combined to account for more variables efficiently.

IV. Part 4

To ensure the best defect detection within the 80-120 test case limit, I would use Boundary Value Analysis (BVA), Equivalence Class Partitioning (ECP), Decision Tables (DT), Pairwise Testing, and Pairwise Orthogonal Arrays (OA) to balance efficiency and coverage. Variable A requires BVA and ECP, testing 6 boundary values and 3 equivalence classes, totaling 9 test cases, since boundary defects are common. Decision Tables (DT) for Variables B, C, and D ensure that all critical rule-based conditions are tested with 12 test cases. To reduce redundancy while ensuring strong interaction coverage, Pairwise Testing (L16 OA for B, C, and D with levels 4×5×2) is applied, requiring 16 test cases. To handle uncommon but high-risk scenarios, I would add 10-15 edge and negative test cases. Finally, Pairwise OA (L_{16}(4 5 2)) ensures full two-way interaction coverage, adding 30-40 more test cases, bringing the total to 80-110 test cases. This strategy maximizes error detection while staying within constraints by covering all critical boundary, decision-based, and interaction-driven defects efficiently.

Reflection:

- What I learned:

    - How to reduce the number of test cases with pairwise testing and orthogonal arrays
    - Understanding mixed-level orthogonal arrays. I learned how to account for variables with multiple different levels.
    - Selecting the correct orthogonal array by evaluating the levels of each variable.
- What worked well:
    - I was able to successfully apply pairwise arrays and orthogonal arrays to reduce redundant test cases.
    - I was also able to correct select the orthogonal arrays for different variable levels.
- Challenges faced:
    - I struggled with handling mixed-level orthogonal arrays, as I had to modify the arrays.
    - Understanding orthogonal arrays, and the decisions required in forming one.

---

5. Honor Pledge

"I pledge my honor that I have abided by the Stevens Honor System."

Signed: *Ryan Davis*
