HW 2 - Pairwise Testing

A controller for an electric kitchen range controls four burners and the oven.

Each burner is controlled by a knob that has settings such as Warm, Low, Medium, and High, but the test team has decided to divide this into three equivalence classes: Off, Regulated, and High. (When the burner is on high, full current goes through the burner, but when the burner is turned on to any other setting, the current is regulated. We assume the current regulator for the burners has been thoroughly tested.)

The oven has four settings: Off, Bake, Broil, and Clean.

The chief of the testing department has decided he wants to use pairwise testing to test the five components, to make sure that no setting on one component interferes with a setting on another component.

This may seem like a hardware test, but, because all of the stove's components are controlled by a programmable microprocessor made by your company, it's really a firmware test, too.

With the aforementioned input values, there are 324 possible configurations is ($3^4 \times 4$). Because each test case takes at least 20 minutes to run (the range burners and oven must reach and maintain the proper temperature for each case, and then cool off before the next case starts), the test team wants to minimize the number of test cases while ensuring all possible pairings are included in the test plan.

Your assignment:

1. Find a pairwise testing tool to generate the orthogonal array. (Links are on Pilot.)
2. Use the tool to get a set of test cases. Make sure your test cases seem reasonable and logical. Tweak them if necessary, but explain what you did.
3. If necessary, translate the output from the orthogonal array generator to something that looks like it would belong in the test plan for your company. (Remember, Alice works there.)

Clarification for #3 (the column on the left is just an example; it's not the only acceptable way to do this):

This is okay:	This is not:
Test Case 1: Burner 1: Off Burner 2: Off Burner 3: High Burner 4: High Oven: Broil	**Orthogonal Array:** 0 0 2 2 2 0 0 2 2 1 : :