Oliver Yu

## How many differences were found initially (before fixing bugs)

The amount of errors before fixing bugs on either of our sides were too many to count. As we later found out due to differences in file structure and input output format there was no way to actually test the files before making sure things were fixed on both ends.

## How many differences were found at the end (after bugs were fixed) / How many bugs were fixed by you and how many were fixed by the person whose code you're testing against

After attempting to fix all the bugs, there were still around 250ish differences out of 10,000 tests, most of which consists of the same error that were mainly due to how my tests were generated. The way I distributed my tests made sure that you would get 100% or close to 100% code coverage over a large number of tests. For every test case there would a equal chance of the input being valid or invalid and based on that decision there are equal chances for the user to get each type or shape or error. This means if one of my error generation was wrong then a good percentage of the differences could be attributed to one bug. One of the persisting bugs on my partner's end was colinear points, where the points lie on the same line. For that error I made several different edge cases and the classifier could not identify them all even after some bug fixing. There is also a bug on my end that I could not figure out which was the generation of interesting lines for error 3. Approximately 2 out of 3 error 3s generated would not have an intersection even though all three edge cases were accounted for. I fixed around 6 bugs around the generation of errors 3 and 4 while my partner had to fix 2-3 for checking colinear points and tried to add more robust code for nullspace transformations such as leading and trailing whitespaces and multiple whitespaces between valid inputs.