(Preliminary) General Observations:
These don't necessarily apply to what every team did on Assignment #5, but they should be taken into account by all teams in improving for Assignment #6 part 2 and beyond.

(1) We should be making fuller use of the API specification to place limitations on the parameters. Possible examples to consider:
    a. Do we want problem #'s that are negative?
    b. Are there any limits on latitude & longitude?
    c. Can the dimensions of an obstacles be negative?
(2) Question: How can we share (if needed) definitions of objects between API definitions?
    a. For example, how can we share the definition of a coordinates?
    b. Or of a problemID?
    c. Does it matter if we use int32 in some places and int64 in others? How can we make that consistent?
(3) While It is easy to put tests together in Python, it doesn't provide much in the way of organization or documentation. How to address that?
(4) The external API testing should move beyond one-time API calls and into sequences of calls that validate that the API is able to retrieve what it stores, that it provides the required consistency model, etc.
(5) Building for modifications:
    a. Definitions of obstacles/boundaries that provide for new types of objects (e.g., not just rectangles) by, for example, naming the object type.
    b. Including API version number to ensure that when a client is using the API, it indicates which version it is using so it gets the correct API.
(6) Was good use made of generating the client SDK from an API definition?
(7) Most of the Python code being written by the teams is pretty bad code. Just because you are writing in a scripting language doesn't mean (a) it has to be bad or (b) you should disregard everything you've learned about programming. Google, for example, makes extensive use of Python, but they don't use it because they want to write bad code more quickly, instead, they use it to write good code quickly (when they use it).
    a. Here is a link to some Google guidelines on Python code: https://www.chromium.org/chromium-os/python-style-guidelines
    b. And please note that those are just **style** guidelines, they aren't really getting much into code design.