Pulumock Assessment Answers

1. **Which approach had a smoother learning curve in regards to unit testing?**

P1	Pulumock
P2	Pulumock
P3	Pulumock
P4	Pulumock

	Motivate your answer
P1	Couldn't and wouldn't have written a test if I had to use the normal Pulumi approach. Needed guiding from tester to even finish the second one in a timely manor.
P2	cleaner documentation, better naming conventions, easier to get started
P3	It was much easier to mock the resources and a smoother experience. The documentation was well written and barely any guidance was needed to create the tests, unlike without Pulumock where I needed additional guidance to even finish the tasks provided.
P4	Writing unit tests with pulumock requires less boilerplate code and seems more intuitive. After writing one test, I immediately understood how to do it.

2. **Which approach offered better readability in your code?**

P1	Pulumock
P2	Pulumock
P3	Pulumock
P4	Pulumock

	Motivate your answer
P1	The builder pattern made it super simple to get an overview.
P2	the only iffy thing is chained build and buildAsync, but otherwise a lot more readable and cleaner
P3	Pulumock made the code much more readable and easier to understand as there wasn't that much "unclear" jargon as when not using Pulumock. There were less steps to finishing the tasks.
P4	There is less boilerplate code and naming for methods, etc. is simpler and more descriptive, also all of the code is close to each other

3. **Which approach offered better reusability in your code?**

P1	Pulumock
P2	Pulumock
P3	Pulumock
P4	Pulumock

Motivate your answer

P1	Builder pattern <3
P2	not really applicable given only 2 tests were written and they touched upon two separate parts of what can be tested, but judging by the way the whole testing class is structured it would be easier to write more tests using Pulumock
P3	I didn't really experience the reusability of the tool as there wasn't enough tasks to implement. But I can see that Pulumock is more reusable than the standard approach.
P4	Few lines with pulumock vs multiple lines without, but the same functionality

4. Which approach would you personally prefer to use in a real-world project?

P1	Pulumock
P2	Pulumock
P3	Pulumock
P4	Pulumock

Motivate your answer

P1	Wouldn't write a Pulumi normal test ever, would test by deploying to an environment instead.
P2	an extra dependency to save hours of headaches, though not having a community around it can suck if you run into a problem, and LLMs won't be able to help you
P3	Pulumock all the way, I would go mad without it.
P4	Easier to learn and easier to modify, also there is documentation with examples.

5. What challenges (if any) did you face when using either approach?

P1	I could barely finish with the normal way at all.
P2	lack of experience with these types of unit testing and dotnet development, so a lot of syntax unclarities
P3	Writing tests without pulumock felt cumbersome, I had hard time understanding how to properly do them
P4	My greatest challenges were that i've never worked in C# or .NET.

6. What was your overall experience using Pulumock?

P1	Great!
P2	loved it!
P3	It was great, the contrast from not using it to using it was stark and if I in the future were to work on a project were I could use Pulumock I would do it in a heartbeat.
P4	Positive, it is easy to use and easy to grasp

7. Do you have any suggestions for improving Pulumock?

P1	Would have to be more into Pulumi to answer this.
P2	not really, it's really sweet, would be neat to get it popular and get a community around it so that it becomes the new standard, because I really can't see any downsides to using it

P3	The Build and BuildAsync were a little bit confusing, but that might be due to inexperience.
P4	no