Unit-Testing 2: Mocking and Stubbing with Dependency Injection

1.

```py
def test_get_country_code():
    test_countries = [
        {"name": "United Kingdom", "alpha3Code": "UK"},
        {"name": "United States", "alpha3Code": "USA"}
    ]

    test_name = "United Kingdom"
    expected = "UK"
    actual = get_country_code(test_countries, test_name)

    assert expected == actual
    print("test_get_country_code: Passed")
```

1.

```py
def test_get_country_currency():
    test_countries = [
        {"name": "United Kingdom", "currencies": [{"code": "GBP"}]},
        {"name": "United States", "currencies": [{"code": "USD"}]}
    ]

    test_name = "United Kingdom"
    expected = "GBP"
    actual = get_country_currency(test_countries, test_name)

    assert expected == actual
    print("test_get_country_currency: Passed")
```

1.

```py
def test_transform():
    def mock_get_countries():
        return [{"name": "United Kingdom", "alpha3Code": "UK",
"currencies": [{"code": "GBP"}]}]
```

```
    test_name = "United Kingdom"
    expected = {"name": "United Kingdom", "country_code": "UK",
"currency_code": "GBP"}
    actual = transform(test_name, mock_get_countries)

    assert actual == expected
    print("test_transform: Passed")
```

1.

```py
def test_show_country_info():

    # Assemble
    expected = {"name": "United Kingdom", "country_code": "UK",
"currency_code": "GBP"}

    def mock_get_countries():
        return [{"name": "United Kingdom", "alpha3Code": "UK",
"currencies": [{"code": "GBP"}]}]

    def mock_print(*args):
        assert args[0] == expected # We're actually asserting here.
        # This may be difficult to test if we had more than one use of
'print'
        # Mocking libraries will help us out with this sort of problem by
providing more features

    def mock_input(msg):
        return 0
    # Act
    # Here we are using actual information of 'transform' method, but we
could also choose to mock it.
    # The test will still use our 'mock_get_countries' as we are already
passing this in to show_country_info
    show_country_info(mock_get_countries, mock_print, mock_input)

    print("test_show_country_info: Passed")
```