Q1

a)

In this case, the bottom up approach would be the best choice.

- Due to the large number of submodules, it would be better to test that these individual modules are working before testing the higher level modules. This would allow more meaningful testing of the higher level modules and allow critical functionality to be tested first.
- The critical functionality is likely at the lowest level. By using the bottom up approach, the critical functionality will be tested multiple times when it is used at higher levels.
- Tests will be easier to create and verify. If we were to consider top-down testing, creating stubs for so many different submodules would be very difficult.
- In this system, it seems like an object-oriented design methodology was used. In general, the bottom up approach is good at dealing with object-oriented programming.

b)

We want to choose the layer that will minimize the number of stubs and drivers. Below is a table for the number of stubs and drivers required when a layer is selected as the target layer (discounting the first and last layers). Let "TD" denote "top down" and "BU" denote "bottom up."

Layer	TD Stubs	TD Drivers	BU Stubs	BU Drivers	Total
2	7	1	0	20	28
3	13	1	0	14	28
4	18	1	0	11	30

We see that layer 2 and 3 result in the least number of total stubs/drivers. To break the tie, we use the fact that, in general, stubs are harder to set up than drivers. As such, we pick the layer with less stubs: layer 2.

Q2

If G is the control flow graph of a program, the McCabe complexity measure states

$$v = e - n + 2 \tag{1}$$

where v is the number of linearly independent paths in G (and also the cyclomatic complexity), e is the number of edges, and n is the number of nodes. As such, we find the cyclomatic complexity of the modules are

$$v(G_A) = (8) - (7) + 2 = 3, \tag{2}$$
$$v(G_B) = (4) - (4) + 2 = 2, \tag{3}$$
$$v(G_C) = (6) - (6) + 2 = 2, \tag{4}$$
$$v(G_D) = (7) - (6) + 2 = 3, \tag{5}$$
$$v(G_E) = (6) - (6) + 2 = 2, \tag{6}$$
$$v(G_F) = (8) - (7) + 2 = 3, \tag{7}$$
$$v(G_G) = (4) - (5) + 2 = 1, \text{ and} \tag{8}$$
$$v(G_H) = (5) - (5) + 2 = 2. \tag{9}$$
$$\tag{10}$$

By applying design reduction, we find the integration complexity of the modules are

$$iv(G_A) = 3, \tag{11}$$
$$iv(G_B) = 1, \tag{12}$$
$$iv(G_C) = 2, \tag{13}$$
$$iv(G_D) = 1, \tag{14}$$
$$iv(G_E) = 2, \tag{15}$$
$$iv(G_F) = 2, \tag{16}$$
$$iv(G_G) = 1, \text{ and} \tag{17}$$
$$iv(G_H) = 1. \tag{18}$$
$$\tag{19}$$

We can calculate the complexity of integration testing S via

$$S = \sum_{i=1}^{n} iv(G_i) - n + 1 \tag{20}$$

where n is the number of modules. Using this, we have

$$S = (3) + (1) + (2) + (1) + (2) + (2) + (1) + (1) - (8) + 1 \tag{21}$$

$$= 6. \tag{22}$$

This means we have 6 independent integration tests. The independent integration tests are

1. $A \longrightarrow B \longleftarrow A$,
2. $A \longrightarrow C \longleftarrow A$,
3. $A \longrightarrow C \longrightarrow D \longleftarrow C \longleftarrow A$,
4. $A \longrightarrow E \longrightarrow A$,
5. $A \longrightarrow E \longrightarrow F \longleftarrow E \longleftarrow A$, and
6. $A \longrightarrow E \longrightarrow F \longrightarrow G \longrightarrow H \longleftarrow G \longleftarrow F \longleftarrow E \longleftarrow A$.

Q3

Let x be the input array. Assume any operations on x apply to all elements of x (for example, $x + 2$ would mean incrementing all values in x by 2). Two metamorphic relations are

$$R_1 = \left\{(x,\ \text{product}(x),\ C \ \mid\ C \in \Re,\ C^{\text{len}(x)}\text{product}(x)) = \text{product}(Cx))\right\} \tag{23}$$

and

$$R_2 = \left\{(x,\ \text{product}(x),\ \mid\ \text{product}(x) = \text{product}(x.\text{reverse}()))\right\}. \tag{24}$$