Answer Key

1. **D**
2. **B**
3. **E**
4. **D**
5. **B**
6. **C**
7. **B**
8. **D**
9. **A**
10. **B**
11. **A**
12. **C**
13. **C**
14. **A**
15. **E**
16. **D**
17. **E**
18. **A**
19. **C**
20. **B**
21. **B**

Answer Explanations

1. **(D)** Tail recursion is when the recursive call of a method is made as the last executable step of the method. Divide-and-conquer algorithms like those used in merge sort or quicksort have recursive calls *before* the last step. Thus, statement II is false.

2. **(B)** Code segment I is wrong because there is no base case. Code segment III is wrong because, besides anything else, `sum(n)` prevents the method from terminating—the base case `n == 1` will not be reached.

3. **(E)** When `stringRecur` is invoked, it calls itself irrespective of the length of `s`. Since there is no action that leads to termination, the method will not terminate until the computer runs out of memory (run-time error).

4. **(D)** The base case is `s.length() ≥ 15`. Since `s` gets longer on each method call, the method will eventually terminate. If the original length of `s` is `≥ 15`, the method will terminate without output on the first call.

5. **(B)** Letting R denote the method `result`, we have

$$\begin{aligned} R(5) &= 2 * R(4) \\ &= 2 * (2 * (R(3))) \\ &= \cdots \\ &= 2 * (2 * (2 * (2 * R(1)))) \\ &= 2^5 \\ &= 32 \end{aligned}$$

6. **(C)** For `result(n)` there will be $(n-1)$ recursive calls before `result(1)`, the base case, is reached. Adding the initial call gives a total of n method calls.

7. **(B)** This method returns the nth term of an arithmetic sequence with first term `a` and common difference `d`. Letting M denote method `mystery`, we have

$$\begin{aligned} M(3,2,6) &= 6 + M(2,2,6) \\ &= 6 + (6 + M(1,2,6)) \quad \text{(base case)} \\ &= 6 + 6 + 2 \\ &= 14 \end{aligned}$$

8. **(D)** Here are the recursive calls that are made, in order: $f(6,8) \rightarrow f(6,2) \rightarrow f(4,2) \rightarrow f(2,2)$, base case. Thus, 2 is returned.