1636

**Review Questions #s 1 - 10 page 1007**

1. B
2. A
3. D
4. C
5. D
6. A
7. True
8. False
9. False
10. False

**Find the Error #1 page 1008**

1. The program will run infinetly

**Algorithm Workbench #1-7 Page 1008-1009**



|  |
| --- |
| public static void display(String s) {  if (s.length() == 0)  return;  System.out.println(s.charAt(0));  display(s.substring(1));  } |

|  |
| --- |
| public static void displayReverse(String s) {  if (s.length() == 0)  return;  displayReverse(s.substring(1));  System.out.println(s.charAt(0));  } |

1. 10

|  |
| --- |
| 0  1  2  3  4  5  6  7  8  9  10 |

1. 55

|  |
| --- |
| public static void sign(int n) {  if (n > 0) {  System.out.println("No Parking");  sign(n - 1);  }  } |

|  |
| --- |
| public static int factorialLoop(int n) {  int fact = 1;  for (; n > 0; fact \*= n--);  return fact;  } |

**Short Answer #1-7 Page 1009**

1. Recursion is simply a method call in which the method being called is the same as the one making the call while iteration is when a loop is repeatedly executed until a certain condition is met.
   1. The condition under which a recursive function returns without calling itself, thereby ending the recursive calls.
   2. A case within a recursive solution that involves reducing the overall problem to a simpler problem of the same kind that can be solved by a recursive call
   3. For question 3 the base case is reached when arg is equal to 10.
   4. For question 4 the base case is reached when arg is equal to 10.
   5. For question 5 the base case is reached when num is less-than or equal to 0.
2. Indirect recursion. There are more function calls to keep up with.
3. Recursive functions are less efficient, due to the overhead associated with each function call.
4. By reducing the problem with each recursive call, the base case will eventually be reached and the recursion will stop.
5. A parameter becomes smaller