

Name: _____

Date: _____

Period: _____

Recursion Practice Worksheet

Instructions: Solve each problem carefully. For free-response questions, show all steps and reasoning. For multiple-choice questions, circle the correct answer. For short-answer questions, write your answer in the space provided.

1. **Multiple Choice:** Consider the following recursive method:

```
public static int mystery(int n) {  
    if (n <= 0) return 0;  
    return n + mystery(n - 2);  
}
```

What is the value of `mystery(7)`?

- (A) 12
- (B) 16
- (C) 20
- (D) 24
- (E) 28

2. **Free Response:** Write a recursive method `int sumDigits(int n)` that returns the sum of the digits of a non-negative integer `n`. For example, `sumDigits(123)` should return $1 + 2 + 3 = 6$. Assume `n` is non-negative. Show your method below.

3. **Short Answer:** The following recursive method is intended to compute x^n for integer $n \geq 0$:

```
public static double power(double x, int n) {  
    if (n == 0) return 1;  
    return x * power(x, n - 1);  
}
```

How many recursive calls are made when computing `power(2, 5)`? (Do not count the initial call.)

4. **Multiple Choice:** What does the following recursive method return for `foo(5)`?

```
public static int foo(int n) {  
    if (n < 2) return n;  
    return foo(n - 1) + foo(n - 2);  
}
```

- (A) 3
- (B) 5
- (C) 8
- (D) 13
- (E) 21

5. **Free Response:** Write a recursive method `boolean isPalindrome(String s)` that returns `true` if the string `s` is a palindrome (reads the same forward and backward), and `false` otherwise. Ignore case. You may assume the string contains only letters. Show your method below.

6. **Short Answer:** Consider the recursive method:

```
public static int g(int n) {  
    if (n < 10) return n;  
    return g(n / 10) + n % 10;  
}
```

What is the value of $g(9876)$?

7. **Multiple Choice:** How many times is the recursive method called (including the initial call) when evaluating `h(4)`?

```
public static int h(int n) {  
    if (n <= 1) return 1;  
    return h(n - 1) + h(n - 2);  
}
```

- (A) 5
- (B) 7
- (C) 9
- (D) 11
- (E) 13

8. **Free Response:** Write a recursive method `void printReverse(String s)` that prints the string `s` in reverse order. For example, `printReverse("hello")` should print `"olleh"`. Show your method below.

9. **Short Answer:** The following recursive method is intended to compute the greatest common divisor (GCD) of two positive integers using Euclid's algorithm:

```
public static int gcd(int a, int b) {  
    if (b == 0) return a;  
    return gcd(b, a % b);  
}
```

What is the value of `gcd(48, 18)`?

10. **Multiple Choice:** What is the output of the following code?

```
public static void main(String[] args) {  
    System.out.println(quiz(4));  
}  
public static int quiz(int n) {  
    if (n == 0) return 0;  
    return n + quiz(n - 1);  
}
```

- (A) 6
- (B) 8
- (C) 10
- (D) 12
- (E) 14

Answer Key

1. **Answer: B (16)**

Working: $\text{mystery}(7) = 7 + \text{mystery}(5) = 7 + 5 + \text{mystery}(3) = 7 + 5 + 3 + \text{mystery}(1) = 7 + 5 + 3 + 1 + \text{mystery}(-1) = 7 + 5 + 3 + 1 + 0 = 16$.

2. **Answer:**

```
public static int sumDigits(int n) {  
    if (n < 10) return n;  
    return n % 10 + sumDigits(n / 10);  
}
```

3. **Answer: 5**

Working: $\text{power}(2,5)$ calls $\text{power}(2,4)$, which calls $\text{power}(2,3)$, which calls $\text{power}(2,2)$, which calls $\text{power}(2,1)$, which calls $\text{power}(2,0)$. That is 5 recursive calls.

4. **Answer: B (5)**

Working: $\text{foo}(5) = \text{foo}(4) + \text{foo}(3) = (\text{foo}(3) + \text{foo}(2)) + (\text{foo}(2) + \text{foo}(1)) = ((\text{foo}(2) + \text{foo}(1)) + (\text{foo}(1) + \text{foo}(0))) + ((\text{foo}(1) + \text{foo}(0)) + 1) = ((1 + 1) + 1 + 0) + (1 + 0 + 1) = 5$.

5. **Answer:**

```
public static boolean isPalindrome(String s) {  
    if (s.length() <= 1) return true;  
    if (s.charAt(0) != s.charAt(s.length()-1)) return false;  
    return isPalindrome(s.substring(1, s.length()-1));  
}
```

6. **Answer: 30**

Working: $g(9876) = g(987) + 6 = (g(98) + 7) + 6 = ((g(9) + 8) + 7) + 6 = (9 + 8) + 7 + 6 = 30$.

7. **Answer: C (9)**

Working: $h(4)$ calls $h(3)$ and $h(2)$. $h(3)$ calls $h(2)$ and $h(1)$. $h(2)$ calls $h(1)$ and $h(0)$. $h(1)$ and $h(0)$ are base cases. Total calls: $h(4)$, $h(3)$, $h(2)$ (twice), $h(1)$ (three times), $h(0)$ (once) = $1+1+2+3+1 = 8$? Wait, recount: $h(4)$ (1), $h(3)$ (2), $h(2)$ from $h(4)$ (3), $h(2)$ from $h(3)$ (4), $h(1)$ from $h(3)$ (5), $h(1)$ from first $h(2)$ (6), $h(0)$ from first $h(2)$ (7), $h(1)$ from second $h(2)$ (8), $h(0)$ from second $h(2)$ (9). So 9 calls.

8. **Answer:**

```
public static void printReverse(String s) {  
    if (s.length() == 0) return;  
    printReverse(s.substring(1));  
    System.out.print(s.charAt(0));  
}
```

9. **Answer: 6**

Working: $\text{gcd}(48, 18) = \text{gcd}(18, 48 \% 18 = 12) = \text{gcd}(12, 18 \% 12 = 6) = \text{gcd}(6, 12 \% 6 = 0) = 6$.

10. **Answer: C (10)**

Working: $\text{quiz}(4) = 4 + \text{quiz}(3) = 4 + 3 + \text{quiz}(2) = 4 + 3 + 2 + \text{quiz}(1) = 4 + 3 + 2 + 1 + \text{quiz}(0) = 4 + 3 + 2 + 1 + 0 = 10$.