

3. Infinite recursion happens either because the parameter values do not get simpler or because a special terminating case is missing.

8. The following code recursively computes a Fibonacci number.  
Insert the missing item for the ? .

```
public static long ? (int n)
{
    if (n <= 2)
        return 1;
    else
        return fib(n - 1) + fib(n - 2);
}
```

9. The recursive call in the following method is factorial(n-1).

```
int factorial(int n)
{
    if(n <= 1)
        return 1;
    else
        return(n * factorial(n-1));
}
```

12. A recursive computation solves a problem by using the solution of the same problem with simpler values. 2) \_\_\_\_\_

15. A common programming error is a(n) \_\_\_\_: a method calling itself over and over with no end in sight.

- A) mutual recursion
- B) permutation
- C) infinite recursion
- D) triangle number

16. The string "eat" has \_\_\_\_ permutations.

- A) two
- B) four
- C) six
- D) eight

**Extra Credit:** (1 pt)

Consider the following method:

```
public static void doSomething(int n)
{
    if (n > 0)
    {
        doSomething(n - 1);
        System.out.print(n);
        doSomething(n - 1);
    }
}
```

What would be output following the call doSomething(3)? EC) \_\_\_\_\_

Questions #14 and #15 refer to method t:

```
//Precondition: n > 1
public static int t(int n)
{
    if (n == 1 || n == 2)
        return 2*n;
    else
        return t(n-1) - t(n-2);
}
```

14. What will be returned by t(5)? 14) \_\_\_\_\_
- A) 4
  - B) 2
  - C) 0
  - D) -2
  - E) -4

15. For the method call t(6), how many calls to t will be made, including the original call? 15) \_\_\_\_\_
- A) 6
  - B) 7
  - C) 11
  - D) 15
  - E) 25

III. Answer the following (1 pt each)

1. What is the terminating condition in the following code? 1) \_\_\_\_\_
- ```
int factorial(int n)
{
    if(n <= 1)
        return 1;
    else
        return(n * factorial(n - 1));
}
```

2. The following code recursively computes a Fibonacci number. Insert the missing item for the ? . 2) \_\_\_\_\_
- ```
public static long int ? (int n)
{
    if (n <= 2)
        return 1;
    else
        return fib(n - 1) + fib(n - 2);
}
```

**Over**

11. Given: 11) \_\_\_\_\_

```
public void printstring (String s)
{
    if (s.length() > 0)
    {
        printString(s.substring(1));
        System.out.print(s.substring(0, 1));
    }
}
```

Which best describes what the printstring method above does?

- A) It prints string s.
- B) It prints string s in reverse order.
- C) It prints only the first character of string s.
- D) It prints only the first two characters of string s.
- E) It prints only the last character of string s.

12. Refer to the method power: 12) \_\_\_\_\_

```
// Precondition: expo is any integer, base is not zero
// Postcondition: base raised to expo power returned
public double power(double base, int expo)
{
    if (expo == 0)
        return 1;
    else
        if (expo > 0)
            return < code >
        else
            return (1/base) * power(base, expo + 1);
}
```

Which < code > correctly completes method power?

- A) (1/base) \* power(base, expo + 1)
- B) (1/base) \* power(base, expo - 1)
- C) base \* power(base, expo + 1)
- D) base \* power(base, expo - 1)
- E) (1/base) \* power(base, expo)

13. Consider method foo; 13) \_\_\_\_\_

```
public int foo(int x)
{
    if (x == 1 || x == 3)
        return x;
    else
        return x * foo(x - 1);
}
```

Assuming no possibility of integer overflow, what will be the value of z after execution of the following statement?

```
int z = foo(foo(3) + foo(4));
```

- A) (15!)/(2!) or 15\*14\*13\*...\*4\*3
- B) 3! + 4! or 3\*2 + 4\*3\*2 or 6 + 24 or 30
- C) (7!)! or 5040!
- D) (3! + 4!)! or (6 + 24)! or 30!
- E) 15

Questions #7 and #8 refer to method result( ):

```
public int result(int n)
{
    if (n == 1)
        return 2 ;
    else
        return 2*result(n - 1);
}
```

7. What value does result(5) return? 7) \_\_\_\_\_  
 A) 64    B) 32    C) 16    D) 8    E) 2

8. If  $n > 0$ , how many times will result( ) be called to evaluate result(n) (including the initial call)? 8) \_\_\_\_\_  
 A) 2    B)  $2^n$     C) n    D) 2n    E)  $n^2$

9. Refer to method f( ) below: 9) \_\_\_\_\_

```
public int f(int k, int n)
{
    if (n == k)
        return k;
    else
        if (n > k)
            return f(k, n - k);
        else
            return f(k - n, n);
}
```

What value is returned by the call f(6, 8)?  
 A) 8    B) 4    C) 3    D) 2    E) 1

10. Given: 10) \_\_\_\_\_

```
public int recur(int[] x, int n) //x is an array of n integers
{
    int t;
    if (n == 1)
        return x[0];
    else
    {
        t = recur(x, n - 1);
        if (x[n - 1] > t)
            return x[n - 1];
        else
            return t;
    }
}
```

What does method recur do?

- A) It finds the largest value in x and leaves x unchanged.
- B) It finds the smallest value in x and leaves x unchanged.
- C) It sorts x in ascending order and returns the largest value in x.
- D) It sorts x in descending order and returns the largest value in x.
- E) It returns x[0] or x[n - 1], whichever is larger.

4. Which of the following, when used as the `< body >` of method `sum`, will enable that method to compute `1 + 2 + . . . + n` correctly for any `n > 0`? 4) \_\_\_\_\_

```
public static int sum(int n)
// Precondition: n > 0
// Postcondition: 1 + 2 + ... + n has been returned
{
    < body >
}
```

```
I    return n + sum(n - 1);
II   if (n == 1)
        return 1;
        else
            return n + sum(n - 1);
III  if (n == 1)
        return 1;
        else
            return sum(n) + sum(n - 1);
```

- A) I only
- B) II only
- C) III only
- D) I and II only
- E) I, II, and III

5. This question refers to methods `f1` and `f2` that are in the same class: 5) \_\_\_\_\_

```
public static int f1(int a, int b)
{
    if (a == b)
        return b;
    else
        return a + f2(a - 1, b);
}
public static int f2(int p, int q)
{
    if (p < q)
        return p + q;
    else
        return p + f1(p - 2, q);
}
```

What value will be returned by a call to `f1(5, 3)`?  
 A) 5    B) 6    C) 7    D) 12    E) 15

6. Refer to method `mystery`: 6) \_\_\_\_\_

```
public int mystery(int n, int a, int d)
{
    if (n == 1)
        return a;
    else
        return d + mystery(n - 1, a, d);
}
```

What value is returned by the call `mystery(3, 2, 6)`?  
 A) 20    B) 14    C) 10    D) 8    E) 2

**I. Answer True or False; use 0 for True and X for False (1 pt each)**

1. Infinite recursion happens either because the parameter values do not get simpler or because a special terminating case is missing. 1) \_\_\_\_\_
2. A recursive computation solves a problem by using the solution of the same problem with more complex values. 2) \_\_\_\_\_
3. The recursive call in the following method is factorial(n - 1). 3) \_\_\_\_\_
 

```

int factorial(int n)
{
    if(n <= 1)
        return 1;
    else
        return(n * factorial(n - 1));
}
      
```

**II. MULTIPLE CHOICE - Choose the best answer (1 pt each)**

1. A common programming error is a(n) \_\_\_\_: a method calling itself over and over with no end in sight. 1) \_\_\_\_\_
  - A) mutual recursion
  - B) permutation
  - C) infinite recursion
  - D) triangle number
2. Which of the following statements about recursion is(are) true? 2) \_\_\_\_\_
  - I Every recursive algorithm can be written using nested loops.
  - II Tail recursion is always used in "divide-and-conquer" algorithms.
  - III In a recursive definition, an object is defined in terms of a simpler case of itself.
  - A) I only
  - B) III only
  - C) I and II only
  - D) I and III only
  - E) II and III only
3. Refer to the method stringRecur: 3) \_\_\_\_\_
 

```

public static void stringRecur(String s)
{
    if (s.length( ) < 15)
        System.out.println(s);
    stringRecur(s + "*");
}
      
```

When will method stringRecur( ) terminate without error?

  - A) Only when the length of the input string is less than 15
  - B) Only when the length of the input string is greater than or equal to 15
  - C) Only when an empty string is input
  - D) For all string inputs
  - E) For no string inputs