Recursive Method Tracing

Consider the following method definition for Questions 1 and 2:

```
public int mystery(int k) {
     if (k == 1) return 0;
     else return(1 + mystery(k/2));
}
```

_____ **1)**

What value is returned by the call `mystery(16)`?

A. 0
B. 2
C. 4
D. 5
E. 16

_____ **2)**

Which of the following best characterizes the values of `k` for which the call `mystery(k)` leads to an infinite recursion?

A. No values
B. All positive values
C. All nonpositive values
D. All odd values
E. All even values

_____ **3)**

Consider the following recursive method. (Assume that method `readInt` reads one integer value typed in by the user.)

```
public static void print(int n) {
    int x;
    if (n > 0) {
        x = readInt();
        if (x > 0) {
            print(n-1);
            System.out.println(x);
        }
        else print(n);

}
```

Which of the following best describes what happens as a result of the call `print(5)`

A. The first five numbers typed by the user are printed in the order in which they are typed.
B. The first five numbers typed by the user are printed in the opposite order to that in which they are typed.
C. The first five positive numbers typed by the user are printed in the order in which they are typed.
D. The first five positive numbers typed by the user are printed in the opposite order to that in which they are typed.
E. Nothing is printed because the call causes an infinite recursion.

_____ **4)**

Consider the following code segment:

```
public void mystery( int j, int k ) {
    if (j != k) mystery( j+1, k );
}
```

Which of the following best characterizes the conditions under which the call `mystery( x, y )` leads to an infinite recursion?

A. All conditions
B. No conditions
C. `x < y`
D. `x > y`
E. `x == y`

Questions 5 & 6 refer to the following code:

```
public static int minMax(int a[], int n)
{
  if (n==1)
    return a[0];
  else
  {
    int x = maxMin(a,n-1);
    if (x < a[n-1])
      return x;
    else
      return a[n-1];
  }
}
```

```
public static int maxMin(int a[], int n)
{
  if (n==1)
    return a[0];
  else
  {
    int x = minMax(a,n-1);
    if (x > a[n-1])
      return x;
    else
      return a[n-1];
  }
}
```

_____ **5)**

What value is printed by the following code?

```
int a[] = {7, 8, 1, 4, 6, 9};
System.out.println(minMax(a, a.length));
```

A. 1
B. 6
C. 7
D. 9
E. None of the above

_____ **6)**

If `b` is an array containing an odd number of integers, how many times (including the initial call) is the method `maxMin` invoked as a result of the code below?

```
minMax(b, b.length)
```

A. `0`
B. `(b.length-1)/2`
C. `(b.length+1)/2`
D. `b.length`
E. Some other number of times