Write down the output for each recursive call on a separate sheet of paper by tracing through it.

[**Exp 1**] What does `someFuncA(4)` output?

```
public static void someFuncA(int n)
{
   if (n <= 1)
   {
      System.out.print(n + " ");
   }
   else
   {
      System.out.print(2*n + " ");
      someFuncA(n-1);
      System.out.print(-3*n + " ");
   }
}
```

[**Exp 2**] Consider the following method.

```
/** @param x an int value such that x >= 0
 */
public void mystery(int x)
{
   System.out.print(x % 10);
   if ((x / 10) != 0)
   {
      mystery(x / 10);
   }
   System.out.print(x % 10);
}
```

Which of the following is printed as a result of the call mystery(1234)?

(a) 1234 (b) 4321 (c) 12344321 (d) 43211234

[**Exp 3**] What does `someFuncB(1)` return?

```
public static int someFuncB(int n)
{
   if (n >=5)
   {
      return 1;
   }
   else
   {
      return 3*someFuncB(n+1);
   }
}
```

[**Exp 4**] What does `someFuncC(5)` return?

```
public static int someFuncC(int n)
{
   if (n == 0 || n == 1)
   {
      return 3;
   }
   else
   {
      return someFuncC(n-1)+2*someFuncC(n-2);
   }
}
```

[**Exp 5**] Consider the following recursive method.

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

Assuming that k is a nonnegative integer and $m = 2^k$, what value is returned as a result of the call `mystery(m)`?

(a) 0 (b) k (c) m (d) (m/2) + 1 (e) (k/2) + 1

[**BJP4 Self-Check 12.3**] What does `mystery1(1), mystery1(4), mystery1(16), mystery1(30)` and `mystery1(100)` output?

```
public void mystery1(int n)
{
    if (n <= 1)
    {
        System.out.print(n);
    }
    else
    {
        mystery1(n / 2);
        System.out.print(", " + n);
    }
}
```

[**BJP4 Self-Check 12.4**] What does `mystery2(113)`, `mystery2(70)`, `mystery2(42)`, `mystery2(30)` and `mystery2(10)` output?

```
public void mystery2(int n)
{
    if (n > 100)
    {
        System.out.print(n);
    }
    else
    {
        mystery2(2 * n);
        System.out.print(", " + n);
    }
}
```

[**BJP4 Self-Check 12.5**] What does `mystery3(0)`, `mystery3(1)`, `mystery3(2)`, `mystery3(4)` and `mystery3(5)` output?

```
public void mystery3(int n)
{
    if (n <= 0)
    {
        System.out.print("*");
    }
    else if (n % 2 == 0)
    {
        System.out.print("(");
        mystery3(n - 1);
        System.out.print(")");
    }
    else
    {
        System.out.print("[");
        mystery3(n - 1);
        System.out.print("]");
    }
}
```

[**BJP4 Self-Check 12.6**] What does `mysteryXY(4, 1)`, `mysteryXY(4, 2)`, `mysteryXY(8, 2)`, `mysteryXY(4, 3)` and `mysteryXY(3, 4)` output?

```
public void mysteryXY(int x, int y)
{
    if (y == 1)
    {
        System.out.print(x);
    }
    else
    {
        System.out.print(x * y + ", ");
        mysteryXY(x, y - 1);
        System.out.print(", " + x * y);
    }
}
```

[**BJP4 Self-Check 12.13**] What does `mystery4(6, 13)`, `mystery4(14, 10)`, `mystery4(37, 10)`, `mystery4(8, 2)` and `mystery4(50, 7)` return?

```
public int mystery4(int x, int y)
{
    if (x < y)
    {
        return x;
    }
    else
    {
        return mystery4(x - y, y);
    }
}
```

[**BJP4 Self-Check 12.14**] What does `mystery5(5, 7)`, `mystery5(12, 9)`, `mystery5(-7, 4)`, `mystery5(-23, -48)` and `mystery5(128, 343)` return?

```
public int mystery5(int x, int y) {
    if (x < 0) {
        return -mystery5(-x, y);
    } else if (y < 0) {
        return -mystery5(x, -y);
    } else if (x == 0 && y == 0) {
        return 0;
    } else {
        return 100 * mystery5(x / 10, y / 10) + 10 * (x % 10) + y % 10;
    }
}
```

[**BJP4 Self-Check 12.15**] What does `mystery6(7, 1)`, `mystery6(4, 2)`, `mystery6(4, 3)`, `mystery6(5, 3)` and `mystery6(5, 4)` return?

```
public int mystery6(int x, int k) {
    if (k == 0 || k == n) {
        return 1;
    } else if (k > n) {
        return 0;
    }     }
    else {
        return mystery6(n - 1, k - 1) + mystery6(n - 1, k);
    }
}
```