30. Consider the following recursive method: _____

```
public static void printStars (int k)
{
     if (k>0)
     {
          printStars(k-1);
          for (int j=1; j<=k; j++)
               System.out.print("*");
          System.out.println();
     }
}
```

What is the output as a result of the call `printStars(4)`?

(A)
```
****
***
**
*
```

(B)
```
*
**
***
****
```

(C)
```
***
**
*
```

(D)
```
*
**
***
```

(E)
```
*
*
*
*
```

31. Consider the following recursive method:

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

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

(A) 0
(B) 2
(C) 4
(D) 5
(E) 16

32. Consider the following recursive method:

```
public static void printArray(String[]a, int k)
{
     if (k < a.length)
     {
          printArray (a, k+1);
          System.out.print(a[k]);
     }
}
```

Assume that array a has been initialized to be of length 4 and to contain the values "a", "b", "c", and "d" (with "a" in a[0], "b" in a[1], and so on.) What is the output as a result of the call `printArray (a, 0)`?

(A) bcd
(B) dcb
(C) abcd
(D) dddd
(E) dcba

33. Questions 33 and 34 refer to the following recursive method:

```
public static int compute (int x, int y)
{
     if (x == y)
          return x;
     else
          return (compute(x+1, y-1));
}
```

What is returned by the call `compute(1,5)`?

(A) 1
(B) 2
(C) 3
(D) 4
(E) No value is returned because infinite recursion occurs.

34. Which of the following calls leads to an infinite recursion?

I. `compute(2,8)`
II. `compute (8,2)`
III. `compute (2,5)`

(A) I only
(B) II only
(C) III only
(D) I and II
(E) II and III

35. 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);
     }
}
```

What is the output of `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 opposite order to that in which they are typed.
(D) The first five positive numbers typed by the user are printed in the order to that in which they are typed.
(E) Nothing is printed because the call causes an infinite recursion.

36. Consider the following method:

```
public void mystery (int a, int b)
{
    System.out.print (a + " ");
    if (a <= b)
        mystery (a  + 5, b -1);
}
```

What is the output when mystery (0, 16) is called?

(A) 0
(B) 0 5
(C) 0 5 10
(D) 0 5 10 15
(E) 0 5 10 15 20

37. What is the output when `smile(4)` is called?

```
public static void smile (int n)
{
    if (n==0)
        return;
    for (int k=1; k<=n; k++)
        System.out.print("smile!");
    smile(n-1);
}
```

(A) `smile!`
(B) `smile!smile!`
(C) `smile!smile!smile!`
(D) `smile!smile!smile!smile!`
(E) `smile!smile!smile!smile!smile!smile!smile!smile!smile!smile!`

38. When `smile(4)` is called, how many times will smile actually be called, including the initial call?

(A) 2
(B) 3
(C) 4
(D) 5
(E) 10

39. Consider the following method:

```
public int getSomething(int value)
{
   if(value < 2)
      return 0;
   else
      return 1 + getSomething(value - 2);
}
```

Assume `val` > 0. What is returned by the call `getSomething(val)`?

(A) `val - 2`
(B) `val % 2`
(C) `(val-1) % 2`
(D) `val / 2`
(E) `(val-1) / 2`

40. Consider the following method:

```
public int change(int value)
{
   if(value < 3)
      return value % 3;
   else
      return value % 3 + 10 * change(value/3);
}
```

What will be returned by the call `change(45)`?

(A) 0
(B) 21
(C) 150
(D) 500
(E) 1200

41. Consider the following method:

```
public void change(int value)
{
    if(value < 5)
        System.out.print("" + value % 5);
    else
    {
        System.out.print("" + value % 5);
        change(value/5);
    }
}
```

What will be printed as a result of the call `change(29)`?

(A) 1
(B) 4
(C) 14
(D) 104
(E) 401

42. Consider the following method:

```
public int getSomething(int value)
{
    if(value < 1)
        return 0;
    else
        return 1 + getSomething(value-1) + getSomething(value-2);
    }
```

What is returned by the call `getSomething(4)`?

(A) 0
(B) 1
(C) 2
(D) 5
(E) 7

43. Consider the following method:

```
public void doSomething(int value)
{
    if(0 < value && value < 10)
    {
        doSomething(value - 1);
        doSomething(value + 1);
        System.out.print(" " + value);
    }
}
```

Which of the following will be printed as a result of the call `doSomething(4)`?

(A) 4 3 2 1 5 6 7 8 9
(B) 4 3 5 2 6 1 7 8 9
(C) 9 8 7 6 5 1 2 3 4
(D) 9 8 7 1 6 2 5 3 4
(E) Nothing will be printed due to an infinite recursion

44. What is the output by the call `fun (3)`?

```
public void fun (int x)
{
    if (x>=1)
    {
        System.out.print(x);
        fun (x-1);
    }
}
```

(A) 3 2 1
(B) 1 2 3
(C) 2 3
(D) 3 2 1 0
(E) Nothing will be printed due to an infinite recursion

45. Consider the following data field and method:

```
private int[] list;

public int getIt(int index)
{
    if(index == list.length - 1)
        return list[index];
    else
    {
        int target = getIt(index + 1);
        if(target < list[index])
          return target;
        else
          return list[index];
    }
}
```

What will be returned by the call `getIt(0)`?

(A) The smallest value in `list`
(B) The index of the smallest value in `list`
(C) The largest value in `list`
. (D) The index of the largest value in `list`
(E) The index of the first occurrence of target in `list`

46. Consider the following data field and method:

```
private int[]list;

public int getIt(int index, int target)
{
      if(index >= list.length)
        return -1;
      else if(target == list[index])
        return index;
      else
        return getIt(index + 1, target);
}
```

What will be returned by the call `getIt(0, 5)`?

(A) The value at index 5 in `list`, or -1 if `list.length` < 5.
(B) The value at index `list.length-1` in `list`, or -1 if `list.length` < 5.
(C) The index of the first occurrence of 5 in `list`, or -1 if 5 does not occur in `list`.
(D) The index of the last occurrence of 5 in `list`, or -1 if 5 does not occur in `list`.
(E) The call will cause an `ArrayIndexOutOfBoundsException`.

47. Consider the following two methods that are declared within the same class:

```
public int supplement(int value)
{
    if(value < 50)
        return reduce(value + 10);
    else
        return value;
}

public int reduce(int value)
{
    if(value > 0)
        return supplement(value – 5);
    else
        return supplement(value);
}
```

What will be returned as a result of the call `supplement(40)`?

(A) 0
(B) -5
(C) 50
(D) 55
(E) Nothing will be returned due to an infinite recursion.

48. Consider the following two methods that are declared within the same class:

```
public int supplement(int value)
{
    if(value < 50)
        return reduce(value + 10);
    else
        return reduce(value);
}

public int reduce(int value)
{
    if(value > 0)
        return supplement(value – 5);
    else
        return value;
}
```

What will be returned as a result of the call `supplement(40)`?

(A) 0
(B) -5
(C) 50
(D) 55
(E) Nothing will be returned due to an infinite recursion.

49. What is the output by the call `fun (3)`?

```
public void fun (int x)
{
    if (x<1)
    {
        System.out.print(x);
    }
    else
    {
        System.out.print(x);
        fun (x-1);
    }
}
```

(A) 3 2 1 0 3 2 1 0
(B) 3 2 1 0
(C) 3 2 1 0 0 1 2 3
(D) 0 1 2 3
(E) Nothing will be printed due to infinite recursion

50. What is the output by the call `fun (3)`?

```
public int fun (int x)
{
    if (x<1)
      return x;
    else
      return x + fun(x-1);
}
```

(A) 3 2 1
(B) 1 2 3
(C) 6
(D) 5
(E) Nothing

51. What is the output by the call `fun (3,6)`?

```
public int fun (int x, int y)
{
    if (y==2)
      return x;
    else
      return fun (x, y-1) + x;
}
```

(A) 3 3 3 3 3
(B) 12
(C) 18
(D) 15
(E) 243

52. Consider the problem of determining the value of an investment (`amt`) that has a given interest rate `(rate)`, compounded annually, after a given period of years `(yrs)`. Each of the following methods correctly computes the value. You may assume all variables have been properly initialized.

```
public double method1 (double amt, int yrs, double rate)
{
    if (yrs >=1)
    for (int y=1; y<=yrs; y++)
        amt += rate*amt;
    return amt;
}

public double method2 (double amt, int yrs, double rate)
{
    if (yrs < 1)
      return amt;
    else
      return method2 (amt, yrs-1, rate) +
           method2 (amt, yrs-1, rate)*rate;
}

public double method3 (double amt, int yrs, double rate)
{
    amt = amt * Math.pow((1+rate), yrs);
}
```

For a large number of years, which statement below best characterizes the execution efficiency of the three code segments?

(A) Method 1 is more efficient than 2 or 3 because it is the most straightforward and understandable method.
(B) Method 2 is more efficient than 1 or 3 because recursion is always the most efficient solution.
(C) Method 3 is more efficient than 1 or 2 because it requires fewer operations.
(D) Methods 1 and 2 are more efficient than 3 because they do not call a method from another class.
(E) Methods 1, 2, and 3 execute equally efficiently.

53. Consider the following recursive method:

```
public static int seq (int x)
{
    if (x<=1 || x==3)
      return x;
    else
      return (seq(x-1) + seq(x-2));
}
```

What value will be printed by the call `seq(5)`?

(A) 1
(B) 3
(C) 4
(D) 7
(E) 11

54. A programmer has mistakenly typed a 2 instead of a 1 in the recursive call in the following search method. What will be the result of starting a search at position 0?

```
// postcondition:   returns first index of key within a at or
//                  after position start
//                  returns -1 if key is not present

public int research (Object [] a, Object key, int start)
{
     if (start == a.length)
     {
          return -1;
     }
     else if (a[start].equals(key))
     {
          return start;
     }
     else
     {
          return research(a, key, start+2);
          // should have been start+1;
     }
}
```

(A) The search will still work, but less efficiently than with the "+1."
(B) The correct value will be returned only when the key is found in an even numbered location.
(C) The correct value will be returned only when the length of the array is even.
(D) An `IndexOutOfBoundsException` will be thrown whenever length of array is odd.
(E) None of these explanations correctly describes when the code will work.

55. Consider the recursive method `minVal` that is intended to return the smallest value among the first `n` values in array a.

```
public static int minVal (int []a, n)
{
     if (n==1)
          return <missing code 1>;
     int min = minVal (a, n-1);
     if (min < a[n-1])
          return <missing code 2>;
     else
          return <missing code 3>;
}
```

Which of the following should be used to complete the three return statements?

	<missing code 1>	*<missing code 2>*	*<missing code 3>*
(A)	a[0]	min	a[n]
(B)	a[0]	a[n]	min
(C)	a[1]	a[min]	a[n-1]
(D)	a[1]	a[min]	a[min-1]
(E)	a[0]	min	a[n-1]

56. Consider the following method:

```
//precondition:  num>=0

public static void mystery (int num)
{
    if (num >1)
        mystery (num/2);
    System.out.print(num%2);
}
```

What is the best postcondition for `mystery`?

(A) Reverses the digits of `num`
(B) Prints the remainder when `num` is divided by 2
(C) Prints one-half `num`
(D) Prints the square root of `num`.
(E) Prints the binary representation of `num`.

57. Which of the following statements about recursive algorithms are true?

I. Recursive algorithms must feature a number as one of their inputs
II. Recursion is best used when there is an identifiable general case and an identifiable simplest case.
III. Some algorithms, such as binary search, require the use of recursion.

(A) I only
(B) II only
(C) III only
(D) Exactly two of the statements are true.
(E) All three of the statements are true.

58. Consider the following method:

```
public void mysteryMix (String str)
{
    int len = str.length();
    if (len >=3)
    {
      mysteryMix (str.substring(0,len/3));
      System.out.print (str.substring(len/3, 2*len/3));
      mysteryMix (str.substring(2*len/3));
    }
}
```

What is the output when `mysteryMix ("la-la-la!")` is called?

(A) `la-la-la!`
(B) `ala-a`
(C) `ala-la-la-l`
(D) `lla-l`
(E) `a-la-a!`

59. Consider the following method:

```
public void mystery (int n)
{
    int i;
    if (n <= 0)
       return;
    for (i=0; i < n; i++)
    {
       System.out.print("-");
    }
    for (i=0; i < n; i++)
    {
       System.out.print("+");
    }
    System.out.print();
    mystery(n-1);                    //recursive call
}
```

What is the output when `mystery (4)` is called?

(A)
```
----++++
```

(B)
```
----++++
----++++
----++++
----++++
```

(C)
```
----+
----++
----+++
----++++
```

(D)
```
-+
--++
---+++
----++++
```

(E)
```
----++++
---+++
--++
-+
```

60. Consider the following method:

```
public void mystery (int n)
{
    int i;
    if (n <= 0)
       return;
    mystery(n-1);                    //recursive call

    for (i=0; i < n; i++)
    {
      System.out.print("-");
    }
    for (i=0; i < n; i++)
    {
      System.out.print("+");
    }
    System.out.print();
}
```

What is the output when `mystery (4)` is called?

(A)
```
----++++
```

(B)
```
----++++
----++++
----++++
----++++
```

(C)
```
----+
----++
----+++
----++++
```

(D)
```
-+
--++
---+++
----++++
```

(E)
```
----++++
---+++
--++
-+
```