14. Given that `n` and `count` are both of type `int`, which statement is true about the following code segments?

```
I  for (count = 1; count <= n; count++)
       System.out.println(count);

II count = 1;
   while (count <= n)
   {
       System.out.println(count);
       count++;
   }
```

(A) I and II are exactly equivalent for all input values `n`.
(B) I and II are exactly equivalent for all input values $n \geq 1$, but differ when $n \leq 0$.
(C) I and II are exactly equivalent only when $n = 0$.
(D) I and II are exactly equivalent only when `n` is even.
(E) I and II are not equivalent for any input values of `n`.

15. The following fragment intends that a user will enter a list of positive integers at the keyboard and terminate the list with a sentinel:

```
int value = 0;
final int SENTINEL = -999;
while (value != SENTINEL)
{
    //code to process value
        ...
    value = IO.readInt();       //read user input
}
```

The fragment is not correct. Which is a true statement?
(A) The sentinel gets processed.
(B) The last nonsentinel value entered in the list fails to get processed.
(C) A poor choice of `SENTINEL` value causes the loop to terminate before all values have been processed.
(D) The code will always process a value that is not on the list.
(E) Entering the `SENTINEL` value as the first value causes a run-time error.

16. Suppose that base-2 (binary) numbers and base-16 (hexadecimal) numbers can be denoted with subscripts, as shown below:

$$2A_{hex} = 101010_{bin}$$

Which is equal to $3D_{hex}$?
(A) 111101_{bin}
(B) 101111_{bin}
(C) 10011_{bin}
(D) 110100_{bin}
(E) 101101_{bin}

17. A common use of hexadecimal numerals is to specify colors on web pages. Every color has a red, green, and blue component. In decimal notation, these are denoted with an ordered triple (x, y, z), where x, y, and z are the three components, each an `int` from 0 to 255. For example, a certain shade of red, whose red, green, and blue components are 238, 9, and 63, is represented as (238, 9, 63).

    In hexadecimal, a color is represented in the format #RRGGBB, where RR, GG, and BB are hex values for the red, green, and blue. Using this notation, the color (238, 9, 63) would be coded as #EE093F.

    Which of the following hex codes represents the color (14, 20, 255)?

    (A) #1418FE
    (B) #0E20FE
    (C) #0E14FF
    (D) #0FE5FE
    (E) #0D14FF

18. In Java, a variable of type `int` is represented internally as a 32-bit signed integer. Suppose that one bit stores the sign, and the other 31 bits store the magnitude of the number in base 2. In this scheme, what is the largest value that can be stored as type `int`?

    (A) 2^{32}
    (B) $2^{32} - 1$
    (C) 2^{31}
    (D) $2^{31} - 1$
    (E) 2^{30}

19. Consider this code segment:

```
int x = 10, y = 0;
while (x > 5)
{
    y = 3;
    while (y < x)
    {
        y *= 2;
        if (y % x == 1)
            y += x;
    }
    x -= 3;
}
System.out.println(x + " " + y);
```

    What will be output after execution of this code segment?

    (A) `1  6`
    (B) `7  12`
    (C) `-3  12`
    (D) `4  12`
    (E) `-3  6`

Questions 20 and 21 refer to the following method, `checkNumber`, which checks the validity of its four-digit integer parameter.

```
//Precondition:  n is a 4-digit integer.
//Postcondition: Returns true if n is valid, false otherwise.
boolean checkNumber(int n)
{
    int d1,d2,d3,checkDigit,nRemaining,rem;
    //strip off digits
    checkDigit = n % 10;
    nRemaining = n / 10;
    d3 = nRemaining % 10;
    nRemaining /= 10;
    d2 = nRemaining % 10;
    nRemaining /= 10;
    d1 = nRemaining % 10;
    //check validity
    rem = (d1 + d2 + d3) % 7;
    return rem == checkDigit;
}
```

A program invokes method `checkNumber` with the statement

```
boolean valid = checkNumber(num);
```

20. Which of the following values of `num` will result in `valid` having a value of `true`?
    (A) `6143`
    (B) `6144`
    (C) `6145`
    (D) `6146`
    (E) `6147`

21. What is the purpose of the local variable `nRemaining`?
    (A) It is not possible to separate `n` into digits without the help of a temporary variable.
    (B) `nRemaining` prevents the parameter `num` from being altered.
    (C) `nRemaining` enhances the readability of the algorithm.
    (D) On exiting the method, the value of `nRemaining` may be reused.
    (E) `nRemaining` is needed as the left-hand side operand for integer division.

22. What output will be produced by this code segment? (Ignore spacing.)

```
for (int i = 5; i >= 1; i--)
{
    for (int j = i; j >= 1; j--)
        System.out.print(2 * j - 1);
    System.out.println();
}
```

(A)
```
9 7 5 3 1
9 7 5 3
9 7 5
9 7
9
```

(B)
```
9 7 5 3 1
7 5 3 1
5 3 1
3 1
1
```

(C)
```
9 7 5 3 1
7 5 3 1 -1
5 3 1 -1 -3
3 1 -1 -3 -5
1 -1 -3 -5 -7
```

(D)
```
1
1 3
1 3 5
1 3 5 7
1 3 5 7 9
```

(E)
```
1 3 5 7 9
1 3 5 7
1 3 5
1 3
1
```

23. Which of the following program fragments will produce this output? (Ignore spacing.)

```
2 - - - - -
- 4 - - - -
- - 6 - - -
- - - 8 - -
- - - - 10 -
- - - - - 12
```

I
```
for (int i = 1; i <= 6; i++)
{
    for (int k = 1; k <= 6; k++)
        if (k == i)
            System.out.print(2 * k);
        else
            System.out.print("-");
    System.out.println();
}
```

II
```
for (int i = 1; i <= 6; i++)
{
    for (int k = 1; k <= i - 1; k++)
        System.out.print("-");
    System.out.print(2 * i);
    for (int k = 1; k <= 6 - i; k++)
        System.out.print("-");
    System.out.println();
}
```

III
```
for (int i = 1; i <= 6; i++)
{
    for (int k = 1; k <= i - 1; k++)
        System.out.print("-");
    System.out.print(2 * i);
    for (int k = i + 1; k <= 6; k++)
        System.out.print("-");
    System.out.println();
}
```

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

24. Consider this program segment:

```
int newNum = 0, temp;
int num = k;            //k is some predefined integer value ≥ 0
while (num > 10)
{
    temp = num % 10;
    num /= 10;
    newNum = newNum * 10 + temp;
}
System.out.print(newNum);
```

Which is a true statement about the segment?

I If `100 ≤ num ≤ 1000` initially, the final value of `newNum` must be in the range `10 ≤ newNum ≤ 100`.
II There is no initial value of `num` that will cause an infinite `while` loop.
III If `num ≤ 10` initially, `newNum` will have a final value of `0`.

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

25. Consider the method `reverse`:

```
//Precondition:  n > 0.
//Postcondition: returns n with its digits reversed.
//Example: If n = 234, method reverse returns 432.
int reverse(int n)
{
    int rem, revNum = 0;

    /* code segment */

    return revNum;
}
```

Which of the following replacements for `/* code segment */` would cause the method to work as intended?

I
```
for (int i = 0; i <= n; i++)
{
    rem = n % 10;
    revNum = revNum * 10 + rem;
    n /= 10;
}
```

II
```
while (n != 0)
{
    rem = n % 10;
    revNum = revNum * 10 + rem;
    n /= 10;
}
```

III
```
for (int i = n; i != 0; i /= 10)
{
    rem = i % 10;
    revNum = revNum * 10 + rem;
}
```

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