SAT PREP

Assignment : AP COMPUTER SCIENCE. TEST

1. What values are stored in x and y after execution of the following program segment?

```
int x = 30, y = 40;
if (x >= 0)
{
    if (x <= 100)
    {
        y = x * 3;
        if (y < 50)
            x /= 10;
    }
    else
        y = x * 2;
}
else
    y = -x;
```

(A) x = 30 y = 90

(B) x = 30 y = -30

(C) x = 30 y = 60

(D) x = 3 y = -3

(E) x = 30 y = 40

2. Suppose that addition and subtraction had higher precedence than multiplication and division. Then the expression

```
2 + 3 * 12 / 7 - 4 + 8
```

would evaluate to which of the following?

(A) 11

(B) 12

(C) 5

(D) 9

(E) −4

3. What value is stored in `result` if

```
int result = 13 - 3 * 6 / 4 % 3;
```

(A) −5

(B) 0

(C) 13

(D) −1

(E) 12

4. 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.

5. Which of the following pairs of declarations will cause an error message?

```
I double x = 14.7;
  int y = x;

II double x = 14.7;
   int y = (int) x;

III int x = 14;
    double y = x;
```

(A) None
(B) I only
(C) II only
(D) III only
(E) I and III only

6. What will the output be for the following poorly formatted program segment, if the input value for num is 22?

```
int num =  call to a method that reads an integer;
if (num > 0)
if (num % 5 == 0)
System.out.println(num);
else System.out.println(num + " is negative");
```

(A) 22
(B) 4
(C) 2 is negative
(D) 22 is negative
(E) Nothing will be output.

7. 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

8. Consider the method `reverse`:

```
/** Precondition:  n > 0.
 *  Postcondition:
 *   - Returns n with its digits reversed.
 *   - Example: If n = 234, method reverse returns 432.
 *  @param n a positive integer
 *  @return n with its digits reversed
 */
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

9. 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
```

10. 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