

Multiple Choice

1. Consider the following code segment:

```
x = !y;  
y = x || y;
```

Assume that `x` and `y` are boolean variables that have been initialized before this code segment executes. What is true about the values of `x` and `y` after the code executes?

- A. `x` is true.
 - B. `y` is true.
 - C. `x` has the same value as it did before the code segment executed.
 - D. `y` has the same value as it did before the code segment executed.
 - E. `y` has the same value as `x` had before the code executed.
2. Assume that `a`, `b`, and `c` are variables of type `int`. Consider the three expressions shown below.

- I. `(a == b) && (a == c) && (b == c)`
- II. `(a == b) || (a == c) || (b == c)`
- III. `((a - b) * (a - c) * (b - c)) == 0`

Assume that subtraction and multiplication never overflow. Which of the expressions will always be true if at least two of `a`, `b`, and `c` are equal?

- A. I only
- B. II only
- C. III only
- D. I and II only
- E. II and III only

3. Given the following methods:

<pre>public static int add(int x, int y) { return x + y; }</pre>	<pre>public static int multiply(int x, int y) { return x * y; }</pre>
--	---

Consider the following expression:

`multiply(2, add(multiply(a, b), add(multiply(a, c), multiply(b, c))))`

Which of the following corresponds to this expression?

- | | |
|------------------------------------|----------------------------------|
| A. $2 * a * b + a * c + b * c$ | D. $a * b + (a * c + b * c)$ |
| B. $a * b + a * c * b * c + 2$ | E. $a * b + (a * c + b * c) * 2$ |
| C. $2 * (a * b + (a * c + b * c))$ | |

4. Consider the following method and select the statement that best describes it.

```
public boolean somethingDifferent(boolean p, boolean q)
{
    return (p || q) && !(p && q);
}
```

- A. `somethingDifferent` always returns false
- B. `somethingDifferent` always returns true
- C. `somethingDifferent` always returns true when `p` is false
- D. `somethingDifferent` always returns true when `q` is false
- E. `somethingDifferent` always returns true when `p` is not equal to `q`

5. Consider the following code segment.

```
int value = 15;
while (value < 28)
{
    value++;
    System.out.println(value);
}
```

What are the first and last values output by the code segment?

	<u>First</u>	<u>Last</u>
A.	15	27
B.	15	28
C.	16	27
D.	16	28
E.	16	29

6. Consider the following method.

```

public void numberCheck(int maxNum)
{
    int typeA =
    0; int typeB
    = 0; int
    typeC = 0;

    for (int k = 1; k <= maxNum; k++)
    {
        if (k % 2 == 0 && k % 5 == 0)
            typeA++;
        if (k % 2 == 0)
            typeB++;
        if (k % 5 == 0)
            typeC++;
    }
    System.out.println(typeA + " " + typeB + " " +
    typeC);
}

```

What is output as a result of the call `numberCheck(50)`?

- A. 5 20 5
 - B. 5 20 10
 - C. 5 25 5
 - D. 5 25 10
 - E. 30 25 10
7. Integers can be represented using different bases. Base 10 (decimal), and base 16 (hexadecimal) are indicated with the subscripts `dec` and `hex`, respectively. For example, the decimal number 23 can also be represented in base 16 as shown below.

$$23_{\text{dec}} = 17_{\text{hex}}$$

Which of the following is equal to $100_{\text{hex}} - 10_{\text{hex}}$?

- A. 15_{dec}
- B. 90_{dec}
- C. 144_{dec}
- D. 240_{dec}
- E. 256_{dec}

Multiple Choice

1. The following method is supposed to calculate and return the sum of the numbers from 1 to n, where $n > 0$.

```
public static int sumToN(int n)
{
    int sum = 0;
    int count = 0;
    while (count < n)
    {
        sum += count;
        count++;
    }
    return sum;
}
```

Which statement best describes what method `sumToN` returns?

- A. The value returned will be the sum of the numbers from 1 to n, inclusive.
- B. The value returned will be the sum of the numbers from 0 to $n - 1$, inclusive.
- C. The value returned will be 0.
- D. The value returned will be n.
- E. No value will be returned because the `while` loop is an infinite loop.

2. Refer to the following methods.

```
public static int add(int x, int y)
{
    return x + y;
}
```

```
public static int multiply(int x, int y)
{
    return x * y;
}
```

What is the value of the expression `multiply(3, add(4, 5))`?

- A. 12
- B. 17
- C. 23
- D. 27
- E. 60

3. Consider the following code segment. What will be printed?

```
String s = "How do you do?";
int index = s.indexOf("o");
while(index >= 0)
{
    System.out.println(index + " ");
    s = s.substring(index + 1);
    index = s.indexOf("o");
}
```

- A. 1 3 2 3
- B. 2 4 3 4
- C. 1 5 8 12
- D. 2 6 9 13
- E. An `IndexOutOfBoundsException` is thrown.

4. Consider the following methods.

```
public void changer(String x, int y)
{
    x = x + "peace";
    y = y * 2;
}
public void test()
{
    String s = "world";
    int n = 6;
    changer(s, n);
    /* End of method */
}
```

When the call `test()` is executed, what are the values of `s` and `n` at the point indicated by `/* End of method */`?

<u>s</u>	<u>n</u>
A. world	6
B. worldpeace	6
C. world	12
D. worldpeace	12
E. peace	12

5. Consider the following class.

```
public class IntCell
{
    private int value;

    public int getValue()
    {
        return value;
    }

    public String toString()
    {
        return "" + value;
    }
}
```

Assume that the following code segment appears in client code.

```
IntCell m = new IntCell();
```

Which of these statements can be used in the client class?

- I. System.out.println(m.getValue());
 - II. System.out.println(m.value);
 - III. System.out.println(m);
-
- A. I only
 - B. II only
 - C. III only
 - D. I and II
 - E. I and III

Multiple Choice

1. Consider the following code segment.

```
String s1 = "poptart";
String s2 = "popcorn";
```

Which of the following expressions can be used to determine if `s1` comes *after* `s2` when ordering strings?

- A. `s1.equals(s2)`
- B. `s1.equals(s2) < 0`
- C. `s1.compareTo(s2)`
- D. `s1.compareTo(s2) > 0`
- E. `s1.compareTo(s2) < 0`

2. Consider the following code segment:

```
public static boolean check(String str)
{
    while (str.length() >= 2)
    {
        int len = str.length();
        if(!str.substring(0, 1).equals(str.substring(len - 1, len)))
            return false;
        else
            str = str.substring(1, len - 1);
    }
    return true;
}
```

Choose the answer that best describes all the cases when method `check` will return `true`.

- A. `str` contains two or more of the same characters
- B. `str` contains two or more of the same characters in a row
- C. `str` starts and ends with the same character
- D. `str` has the same characters in the same order forward as it does backward
- E. `str` contains only two characters

3. Given the following code:

```
String s1 = new String("pickle");
String s2 = new String("pickled peppers");
String s3 = s2.substring(0, 4);
String s4 = new String("pick");
```

Which of the following would return true?

- I. s1.equals(s3)
 - II. s1.compareTo(s4) >= 0
 - III. s3.equals(s4)
- | | | |
|-------------|--------------------|-------------------|
| A. I only | C. I and III only | E. I, II, and III |
| B. III only | D. II and III only | |
4. What is the output from the following code?

```
String s = "Computer Science is fun! ";
String s1 = s.substring(0, 8);
String s2 = s1.substring(2);
String s3 = s2.substring(0, 3);
System.out.println(s3);
```

- | | | | | |
|-------|--------|---------|--------|-------|
| A. mp | B. mpu | C. mput | D. omp | E. om |
|-------|--------|---------|--------|-------|
5. In Java, which class or classes have a `toString` method?

- | | |
|-----------------------------------|--|
| A. Only the Object class | D. Only classes defined by programmers |
| B. Only the String class | E. All Java classes |
| C. Only the built-in Java classes | |

6. Consider the following code segment.

```
String str = "apple-dapple";
int k = str.indexOf("p");
while (k != -1)
{
    str = str.substring(0, k) + "z" + str.substring(k + 1);
    k = str.indexOf("p");
}
System.out.println(str);
```

What is the output when this code segment is executed?

- | | |
|-----------------|-----------------|
| A. apple-dapple | D. azzle-dazzle |
| B. apzle-dapzle | E. ad |
| C. azple-dazple | |

Multiple Choice

1. How can you distinguish a constructor from the methods in a class?
 - I. Constructors have the keyword public.
 - II. Constructors have no return type.
 - III. Constructors have the same name as the class
 - A. I only
 - B. II only
 - C. III only
 - D. II and III only
 - E. I, II, and III
2. The process of creating a new object is called:
 - A. encapsulation
 - B. class
 - C. instantiation
 - D. new
 - E. initialization
3. An object is
 - A. the same as a class.
 - B. an instance of a class
 - C. a primitive type variable.
 - D. a method that accesses class attributes.
 - E. only used as a private instance variable.
4. Class methods that can change an object's state are called:
 - A. accessors
 - B. constructors
 - C. mutators
 - D. methods
 - E. void methods

Given the following incomplete class:

```
public class Binky
{
    //private instance variables not shown
    public Binky() {...}
    public Binky(String name) {...}
    public Binky(String name, int age) {...}

    public String getName() {...}
    public void setName(String name) {...}

    public int getAge() {...}
    public void setAge(int age) {...}

    public String toString() {...}

    //other methods not shown
}
```

5. Which one of the following statements would create a new Binky object using the one parameter constructor?
 - A. Binky baby = Binky("Sally");
 - B. Binky baby = new Binky("Sally");
 - C. Binky = new Binky("Sally");
 - D. baby = Binky("Sally");
 - E. none of these
6. Which of the following statements would instantiate a new Binky object with a given name "Linus" and an age of 0?
 - A. Binky baby = new Binky();
 - B. Binky baby = new Binky();
 baby.setName("Linus");
 baby.setAge(0);
 - C. Binky baby = new Binky("Linus");
 baby.setAge(0);
 - D. Binky baby = new Binky("Linus", 0);
 - E. none of these

7. Given that baby has been instantiated as a Binky object, which of the following statements would be an appropriate call to the getName method?
 - A. Binky.getName();
 - B. baby.getName();
 - C. baby.getName("Lucy");
 - D. System.out.println(baby.getName());
 - E. baby = getName();
8. Given that baby has been instantiated as a Binky object, which of the following statements would be an appropriate call to the setName method?
 - A. Binky.setName("Lucy");
 - B. baby.setName();
 - C. baby.setName("Lucy");
 - D. System.out.println(baby.setName("Lucy"));
 - E. baby = setName("Lucy");
9. Assume that babyBoy and babyGirl have been instantiated as Binky objects. Which of the following code segments would calculate the total of their ages?
 - I. int ageTotal = babyBoy + babyGirl;
 - II. int ageTotal = babyBoy.getAge();
ageTotal += babyGirl.getAge();
 - III. int ageTotal;
ageTotal = babyBoy.getAge() + babyGirl.getAge();
 - A. I only
 - B. II only
 - C. III only
 - D. I and II
 - E. II and III

10. Consider the following classes that will be used to represent points in the xy-coordinate plane.

```

public class Point {
    private int x; // coordinates
    private int y;
    public Point() {
        x = 0;
        y = 0;
    }
    public Point(int a, int b){
        x = a;
        y = b;
    }
    // ... other methods not shown
}

public class NamedPoint extends Point {
    private String name;
    //constructors go here
    //... other methods not shown
}

```

Consider the following proposed constructors for the NamedPoint class.

- I. `public NamedPoint()`
`{`
 `name = "";`
`}`
- II. `public NamedPoint(int d1, int d2, String pointName)`
`{`
 `x = d1; y = d2;`
 `name = pointName;`
`}`
- III. `public NamedPoint(int d1, int d2, String pointName)`
`{`
 `super(d1, d2); name = pointName;`
`}`

Which of these constructors would be legal for the NamedPoint class?

- A. I only
- B. II only
- C. III only
- D. I and III
- E. II and III

Multiple Choice

1. Consider the following instance variable and incomplete method. The method calcTotal is intended to return the sum of all values in vals.

```
private int[] vals;  
  
public int calcTotal()  
{  
    int total = 0;  
  
    /* missing code */  
  
    return total;  
}
```

Which of the code segments shown below can be used to replace `/* missing code */` so that calcTotal will work as intended?

- I.

```
for (int pos = 0; pos < vals.length; pos++)  
{  
    total += vals[pos];  
}
```
 - II.

```
for (int pos = vals.length; pos > 0; pos--)  
{  
    total += vals[pos] ;  
}
```
 - III.

```
int pos = 0;  
while (pos < vals.length)  
{  
    total += vals[pos]; pos++;  
}
```
-
- A. I only
 - B. II only
 - C. III only
 - D. I and III
 - E. II and III

2. What is the output of the following program segment?

```
int[] list = {1, 2, 3, 4, 5, 6, 7, 8, 9};
for (int number : list)
    number *= 10;
for (int number : list)
    System.out.print(number + " ");
A. 1 2 3 4 5 6 7 8 9
B. 10 20 30 40 50 60 70 80 90
C. 10 10 10 10 10 10 10 10 10
D. 11 12 12 14 15 16 17 18 19
E. Runtime Exception Error
```

3. Consider the following instance variable and method.

```
private int[] numbers;

/** Precondition: numbers contains int values in no particular order. */
public int mystery(int num)
{
    for (int k = numbers.length - 1; k >= 0; k--)
    {
        if (numbers[k] < num)
        {
            return k;
        }
    }
    return -1;
}
```

Which of the following best describes the contents of numbers after the following statement has been executed?

- ```
int m = mystery(n);
```
- A. All values in positions 0 through m are less than n.
  - B. All values in positions m+1 through numbers.length-1 are less than n.
  - C. All values in positions m+1 through numbers.length-1 are  $\geq n$ .
  - D. The smallest value is at position m.
  - E. The largest value that is smaller than n is at position m.

4. Consider the following two methods that appear within a single class.

```
public void changeIt(int[] list, int num)
{
 list = new int[5];
 num = 0;

 for (int j = 0; j < list.length; j++)
 {
 list[j] = 0;
 }
}

public void start()
{
 int[] nums = {1, 2, 3, 4, 5};
 int value = 6;

 changeIt(nums, value);

 for (int k = 0; k < nums.length; k++)
 {
 System.out.print(nums[k] + " ");
 }

 System.out.print(value);
}
```

What is printed as a result of the call `start()` ?

- A. 0 0 0 0 0
- B. 0 0 0 0 6
- C. 1 2 3 4 5 6
- D. 1 2 3 4 5 0
- E. `changeIt` will throw an exception.

**Multiple Choice**

1. Consider the following code segment:

```
ArrayList<String> items = new ArrayList<String>();
items.add("A");
items.add("B");
items.add("C");
items.add(0, "D");
items.remove(3);
items.add(0, "E");
System.out.println(items);
```

What is printed as a result of executing the code segment?

- A. [A, B, C, E]
- B. [A, B, D, E]
- C. [E, D, A, B]
- D. [E, D, A, C]
- E. [E, D, C, B]

2. Consider the following instance variable and method.

```
private ArrayList<Integer> list;

public void mystery(int n)
{
 for (int k = 0; k < n; k++)
 {
 Integer num = list.remove(0);
 list.add(num);
 }
}
```

Assume that list has been initialized with the following Integer objects.

[12, 9, 7, 8, 4, 3, 6, 11, 1]

Which of the following represents the list as a result of a call to mystery(3)?

- A. [12, 9, 8, 4, 3, 6, 11, 1, 7]
- B. [12, 9, 7, 8, 4, 6, 11, 1, 3]
- C. [12, 9, 7, 4, 3, 6, 11, 1, 8]
- D. [8, 4, 3, 6, 11, 1, 12, 9, 7]
- E. [1, 11, 6, 12, 9, 7, 8, 4, 3]

16

3. Consider the following method which is intended to remove all of the odd values in a list.

```
public static void removeOdd(List<Integer> allNums)
{
 for (int i = 0; i < allNums.size(); i++)
 {
 if (allNums.get(i).intValue() % 2 == 1)
 {
 allNums.remove(i);
 }
 }
}
```

Which of the code segments shown below will build a list that makes `removeOdd` appear to work as intended?

I. `List<Integer> list = new ArrayList<Integer>();  
for (int i = 0; i < 12; i++)  
{  
 list.add(new Integer(i));  
}`

II. `List<Integer> list = new ArrayList<Integer>();  
for (int i = 0; i < 12; i += 2)  
{  
 list.add(new Integer(i));  
}`

III. `List<Integer> list = new ArrayList<Integer>();  
for (int i = 1; i < 12; i +=2)  
{  
 list.add(i);  
}`

- A. I only
- B. II only
- C. III only
- D. I and II
- E. II and III

4. Consider the following declaration of the class NumSequence, which has a constructor that is intended to initialize the instance variable seq to an ArrayList of numberOfValues random floating-point values in the range [0.0, 1.0).

```
public class numSequence
{
 private List<Double> seq;

 // precondition: numberOfValues > 0
 // postcondition: seq has been initialized to an ArrayList of
 // numberOfValues elements; each contains a random
 // Double value in the range [0.0, 1.0)
 public NumSequence(int numberOfValues)
 {
 /* missing code */
 }
}
```

Which of these code segments could be used to replace */\* missing code \*/* so that the constructor will work as intended?

- I. 

```
List<Double> seq = new ArrayList<Double>();
for (int i = 0; i < numberOfValues; i++)
{
 seq.add(new Double(Math.random()));
}
```
  - II. 

```
seq = new ArrayList<Double>();
for (int i = 0; i < numberOfValues; i++)
{
 seq.add(new Double(Math.random()));
}
```
  - III. 

```
ArrayList<Double> temp = new ArrayList<Double>();
for(int i = 0; i < numberOfValues; i++)
{
 temp.add(new Double(Math.random()));
}
seq = temp;
```
- |             |                   |                    |
|-------------|-------------------|--------------------|
| A. II only  | C. I and II only  | E. II and III only |
| B. III only | D. I and III only |                    |

**Multiple Choice**

1. Consider the following method.

```
public static void mystery(List<Integer> nums)
{
 for (int k = 0; k < nums.size(); k++)
 {
 if (nums.get(k).intValue() == 0)
 {
 nums.remove(k);
 }
 }
}
```

Assume that `List<Integer>` values initially contains the following `Integer` values.

[0, 0, 4, 2, 5, 0, 3, 0]

What will values contain as a result of executing `mystery(values)` ?

- A. [0, 0, 4, 2, 5, 0, 3, 0]
- B. [4, 2, 5, 3]
- C. [0, 0, 0, 0, 4, 2, 5, 3]
- D. [0, 4, 2, 5, 3]
- E. The code throws an `ArrayIndexOutOfBoundsException` exception.

2. Consider the following instance variable and method.

```
private int[] array;

// Precondition: array.length > 0
public int checkArray()
{
 int loc = array.length / 2;
 for (int k = 0; k < array.length; k++)
 {
 if (array[k] > array[loc])
 {
 loc = k;
 }
 }
 return loc;
}
```

Which of the following is the best postcondition for checkArray?

- A. Returns the index of the first element in array array whose value is greater than array[loc]
- B. Returns the index of the last element in array array whose value is greater than array[loc]
- C. Returns the largest value in array array
- D. Returns the index of the largest value in array array
- E. Returns the index of the largest value in the second half of array array

*Questions 3 and 4 refer to the following information.*

Consider the following data field and method. The method `removeDups` is intended to remove all adjacent duplicate numbers from `myData`, but does not work as intended.

```
private ArrayList<Integer> myData;

public void removeDups()
{
 int k = 1;
 while (k < myData.size())
 {
 if (myData.get(k).equals(myData.get(k - 1)))
 {
 myData.remove(k);
 }
 k++;
 }
}
```

For example, if `myData` has the values 3 3 4 4 4 8 7 7 7, after calling `removeDups`, `myData` should have the values 3 4 8 7.

3. Assume that `myData` has the following values: 2 7 5 5 5 5 6 6 3 3 3

Which of the following represents `myData` after the incorrect `removeDups` is executed?

- A. 2 7 5 6 3
  - B. 2 7 5 6 3 3
  - C. 2 7 5 5 6 3 3
  - D. 2 7 7 5 5 5 6 3 3
  - E. 2 7 5 5 5 5 6 6 3 3
4. Which of the following best describes how to fix the error so that `removeDups` works as intended?
- A. `k` should be initialized to 0 at the beginning of the method
  - B. The `while` condition should be `(k < myData.size() - 1)`
  - C. The `if` test should be `(myData.get(k).equals(myData.get(k + 1)))`
  - D. The body of the `if` statement should be `myData.remove(k - 1);`
  - E. There should be an `else` before the statement `k++;`

**Multiple Choice**

1. Consider the following declarations.

```
public class Example0
{
 public void doNothing(Example1 b, Example2 c)
 {
 }
}

public class Example1 extends Example0
{
}

public class Example2 extends Example1
{
}
```

The following initializations appear in a different class.

```
Example0 e0 = new Example0();
Example1 e1 = new Example1();
Example2 e2 = new Example2();
```

Which of the following is a correct call to doNothing?

- A. e0.doNothing(e0, e0);
- B. e1.doNothing(e1, e1);
- C. e1.doNothing(e2, e1);
- D. e2.doNothing(e0, e0);
- E. e2.doNothing(e2, e2);

2. Which of the following statements about a class that contains an abstract method is/are true?

- I. The class must be an abstract class.
  - II. The class may not declare a constructor.
  - III. The class may not be instantiated.
- 
- A. I only
  - B. I and II only
  - C. I and III only
  - D. II and III only
  - E. I, II, and III

3. A bear is an animal, and a zoo contains many animals, including bears. Three classes, Animal, Bear, and Zoo are declared to represent animal, bear, and zoo objects. Which of the following is the most appropriate set of declarations?

A. public class Animal extends Bear  
{  
    ...

}  
public class Zoo  
{  
    private Animal[] zooAnimals;  
}

B. public class Bear extends Animal  
{  
    ...

}  
public class Zoo  
{  
    private Animal[] zooAnimals;  
}

C. public class Animal extends Zoo  
{  
    private Bear myBear;  
}

D. public class Bear extends Animal, Zoo  
{  
    ...

E. public class Bear extends Animal implements Zoo  
{  
    ...

4. Consider the following two classes.

```
public class Dog
{
 public void act()
 {
 System.out.print("run "); eat();
 }
 public void eat()
 {
 System.out.print("eat ");
 }
}

public class UnderDog extends Dog
{
 public void act()
 {
 super.act();
 System.out.print("sleep ");
 }
 public void eat()
 {
 super.eat();
 System.out.print("bark ");
 }
}
```

Assume that the following declaration appears in a class other than Dog.

```
Dog fido = new UnderDog();
```

What is printed as a result of the call fido.act()?

- A. run eat
- B. run eat sleep
- C. run eat sleep bark
- D. run eat bark sleep
- E. Nothing is printed due to infinite recursion.

5. Consider the following declarations

```
public interface Shape
{
 int isLargerThan(Shape other);
 // Other methods not shown
}

public class Circle implements Shape
{
 // Other methods not shown
}
```

Which of the following method headings of `isLargerThan` can be added to the declaration of the `Circle` class so that it will satisfy the `Shape` interface?

- I. public int isLargerThan(Shape other)
  - II. public int isLargerThan(Circle other)
  - III. public boolean isLargerThan(Object other)
- 
- A. I only
  - B. II only
  - C. III only
  - D. I and II only
  - E. I, II, and III

**Multiple Choice**

1. Which of the following is a reason to use an `ArrayList` instead of an array?
  - A. An `ArrayList` allows faster access to its  $k$ th item than an array does.
  - B. An `ArrayList` always uses less memory than an array does.
  - C. An `ArrayList` can store objects and an array can only store primitive types.
  - D. An `ArrayList` resizes itself as necessary when items are added, but an array does not.
  - E. An `ArrayList` provides access to the number of items it stores, but an array does not.
  
2. Consider the following two data structures for storing several million words.
  - I. An array of words, not in any particular order
  - II. An array of words, sorted in alphabetical order

Which of the following statements most accurately describes the time needed for operations on these data structures?

- A. Inserting a word is faster in II than in I.
  - B. Finding a given word is faster in I than in II.
  - C. Finding a given word is faster in II than in I.
  - D. Finding the longest word is faster in II than in I.
  - E. Finding the first word in alphabetical order is faster in I than in II.
- 
3. Consider the following code segment.

```
int[] a;
a = initialize(); // instantiates a and initializes its elements
for (int k = 0; k < a.length; k++)
{
 int pos1 = k;
 int pos2 = a.length - k - 1; swap(a, pos1, pos2);
}
```

Assume method `swap` takes the array `a` and interchanges the values at locations `pos1` and `pos2`. Which of the following best characterizes the effect of the `for` loop?

- A. It sorts the elements of `a`.
- B. It reverses the elements of `a`.
- C. It reverses the order of the first half of `a` and leaves the second half unchanged.
- D. It reverses the order of the second half of `a` and leaves the first half unchanged.
- E. It leaves all of the elements of `a` in their original order.

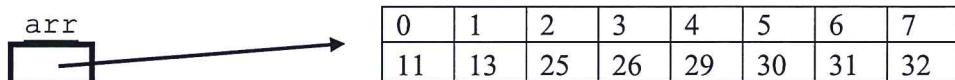
4. Assume that an array contains 100 integers sorted in increasing order. Two alternatives to search the array for a particular integer are sequential and binary search. When searching for a value that is in the array, which of the following best characterizes that greatest number of items in the array that will be examined during each kind of search?

|    | <u>Sequential</u> | <u>Binary</u> |
|----|-------------------|---------------|
| A. | 100               | 1             |
| B. | 100               | 7             |
| C. | 100               | 50            |
| D. | 50                | 7             |
| E. | 50                | 25            |

5. Consider the following method.

```
/** Precondition: arr contains no duplicates;
 * the elements in arr are in sorted order
 * 0 < low < arr.length
 * low - 1 < high < arr.length
 */
public static int mystery(int[] arr, int low, int high, int num)
{
 int mid = (low + high) / 2;
 if (low > high)
 return low;
 else if (arr[mid] < num)
 return mystery(arr, mid + 1, high, num);
 else if (arr[mid] > num)
 return mystery(arr, low, mid - 1, num);
 else // arr[mid] == num
 return mid;
}
```

How many calls to mystery (including the initial call) are made as a result of the call `mystery(arr, 0, arr.length-1, 14)` if arr references the following array?



- A. 1
- B. 2
- C. 4
- D. 7
- E. 8

6. The following `sort` method correctly sorts the integers in `elements` into ascending order.

```

Line 1: public static void sort(int[] elements) {
Line 2: for (int j = 0; j < elements.length - 1; j++) {
Line 3: int index = j;
Line 4: for (int k = j + 1; k < elements.length; k++) {
Line 5: if (elements[k] < elements[index])
Line 6: index = k;
Line 7: }
Line 8: int temp = elements[j];
Line 9: elements[j] = elements[index];
Line 10: elements[index] = temp;
Line 11: }
Line 12: }
```

Which of the following changes to the `sort` method would correctly sort the integers in `elements` into **descending** order?

- I. Replace line 5 with: Line 5: `if (elements[k] > elements[index])`
- II. Replace lines 8-10 with:
 

```

Line 8: int temp = elements[index];
Line 9: elements[index] = elements[j];
Line 10: elements[j] = temp;
```
- III. Replace line 2 with:
 

```
Line 2: for (int j = elements.length - 1; j > 0; j--) {
```

and replace line 4 with:

```
Line 4: for (int k = 0; k < j; k++) {
```

  - A. I only
  - B. II only
  - C. I and II only
  - D. I and III only
  - E. I, II, and III

**Multiple Choice**

1. Consider the following instance variable and method.

```
private int[][] mat;

public void mystery()
{
 for (int row = 1; row < mat.length; row++)
 {
 for (int col = 0; col < mat[0].length; col++)
 {
 if (row != col)
 {
 mat[row][col] = mat[row-1][col];
 }
 }
 }
}
```

Assume that `mat` contains the following values. Note that `mat[0][4]` is 2.

|   |   |   |   |   |
|---|---|---|---|---|
| 4 | 1 | 4 | 3 | 2 |
| 1 | 8 | 7 | 5 | 3 |
| 7 | 4 | 6 | 9 | 2 |
| 3 | 8 | 1 | 2 | 4 |
| 5 | 6 | 7 | 0 | 3 |

What values does `mat` contain after a call to `mystery`?

A.

|   |   |   |   |   |
|---|---|---|---|---|
| 4 | 1 | 4 | 3 | 2 |
| 4 | 8 | 3 | 4 | 2 |
| 4 | 8 | 6 | 4 | 2 |
| 4 | 8 | 6 | 2 | 2 |
| 4 | 8 | 6 | 2 | 3 |

B.

|   |   |   |   |   |
|---|---|---|---|---|
| 4 | 1 | 3 | 4 | 2 |
| 4 | 1 | 3 | 4 | 2 |
| 4 | 1 | 3 | 4 | 2 |
| 4 | 1 | 3 | 4 | 2 |
| 4 | 1 | 3 | 4 | 2 |

C.

|   |   |   |   |   |
|---|---|---|---|---|
| 4 | 1 | 4 | 3 | 2 |
| 4 | 1 | 3 | 4 | 2 |
| 1 | 8 | 7 | 5 | 3 |
| 7 | 4 | 6 | 9 | 2 |
| 3 | 8 | 1 | 2 | 4 |

D.

|   |   |   |   |   |
|---|---|---|---|---|
| 4 | 4 | 4 | 4 | 4 |
| 1 | 1 | 1 | 1 | 1 |
| 7 | 7 | 7 | 7 | 7 |
| 3 | 3 | 3 | 3 | 3 |
| 5 | 5 | 5 | 5 | 5 |

E.

|   |   |   |   |   |
|---|---|---|---|---|
| 4 | 8 | 6 | 2 | 3 |
| 4 | 8 | 6 | 2 | 3 |
| 4 | 8 | 6 | 2 | 3 |
| 4 | 8 | 6 | 2 | 3 |
| 4 | 8 | 6 | 2 | 3 |

2. Consider the following code segment.

```
int[][] mat = new int[3][4];
for (int row = 0; row < mat.length; row++)
{
 for (int col = 0; col < mat[0].length; col++)
 {
 if (row < col)
 {
 mat[row][col] = 1;
 }
 else if (row == col)
 {
 mat[row][col] = 2;
 }
 else
 {
 mat[row][col] = 3;
 }
 }
}
```

What are the contents of `mat` after the code segment has been executed?

- A.  `{{2, 1, 1},  
 {3, 2, 1},  
 {3, 3, 2},  
 {3, 3, 3}}`
- B.  `{{2, 3, 3},  
 {1, 2, 3},  
 {1, 1, 2},  
 {1, 1, 1}}`
- C.  `{{2, 3, 3, 3},  
 {1, 2, 3, 3},  
 {1, 1, 2, 3}}`
- D.  `{{2, 1, 1, 1},  
 {3, 2, 1, 1},  
 {3, 3, 2, 1}}`
- E.  `{{1, 1, 1, 1},  
 {2, 2, 2, 2},  
 {3, 3, 3, 3}}`

3. Consider the following Util class, which contains two methods. The completed sum1D method returns the sum of all the elements of the 1-dimensional array a. The incomplete sum2D method is intended to return the sum of all the elements of the 2-dimensional array m.

```
public class Util
{
 /** Returns the sum of the elements of the 1-dimensional array a */
 public static int sum1D(int[] a)
 { /* implementation not shown */ }

 /** Returns the sum of the elements of the 2-dimensional array m */
 public static int sum2D(int[][] m)
 {
 int sum = 0;

 /* missing code */

 return sum;
 }
}
```

Assume that sum1D works correctly. Which of the following can replace */\*missing code \*/* so that the sum2D method works correctly?

- I. `for (int k = 0; k < m.length; k++) {  
 sum += sum1D(m[k]);  
}`
- II. `for (int[] row : m) {  
 sum += sum1D(row);  
}`
- III. `for (int[] row : m) {  
 for (int v : row) {  
 sum += v;  
 }  
}`

- A. I only
- B. II only
- C. I and II only
- D. II and III only
- E. I, II, and III

4. Consider the following code segment.

```
int[] oldArray = {1, 2, 3, 4, 5, 6, 7, 8, 9};
int[][] newArray = new int[3][3];
int row = 0;
int col = 0;
for (int index = oldArray.length - 1; index >= 0; index--)
{
 newArray[row][col] = oldArray[index];
 col++;
 if (col == newArray[0].length)
 {
 col = 0;
 row++;
 }
}
System.out.println(newArray[0][2]);
```

What is printed as a result of executing the code segment?

- A. 3
- B. 4
- C. 5
- D. 7
- E. 8