test3v2

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]

test3v2

2. Assume mat is defined as follows.

```
int dim = 4;

int[][] mat = new int[dim][dim];
```

Consider the following code segment.

```
int sum = 0;

for (int row = 0; row < dim; row++)

{

  sum = sum + mat[row][dim - 1];

}
```

Assume that mat contains the following values before the code segment is executed. Note that mat[0][3] is 2.

	0	**1**	**2**	**3**
0	1	1	2	2
1	1	2	2	4
2	1	3	2	6
3	1	4	2	8

What value will sum contain after the code segment is executed?

(A) 6

(B) 8

(C) 13

(D) 15

(E) 20

3. 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

4. The code segment below is intended to print the length of the shortest string in the array `wordArray`. Assume that `wordArray` contains at least one element.

```
int shortest = /* missing value */;
for (String word : wordArray)
{
     if (word.length() < shortest)
     {
         shortest = word.length();
     }
}
System.out.println(shortest);
```

Which of the following should be used as the initial value assigned to `shortest` so that the code segment works as intended?

(A) `Integer.MAX_VALUE`
(B) `Integer.MIN_VALUE`
(C) `0`
(D) `word.length()`
(E) `wordArray.length`

5. The following incomplete method is intended to sort its array parameter arr in increasing order.

```
// postcondition: arr is sorted in increasing order
public static void sortArray(int[] arr)
{
  int j, k;

  for (j = arr.length - 1; j > 0; j--)
  {
    int pos = j;

    for ( /* missing code */ )
    {
      if (arr[k] > arr[pos])
      {
        pos = k;
      }
    }
    swap(arr, j, pos);
  }
}
```

Assume that swap(arr, j, pos) exchanges the values of arr[j] and arr[pos]. Which of the following could be used to replace /* *missing code* */ so that executing the code segment sorts the values in array arr?

test3v2

(A) k = j - 1; k > 0; k--
(B) k = j - 1; k >= 0; k--
(C) k = 1; k < arr.length; k++
(D) k = 1; k > arr.length; k++
(E) k = 0; k <= arr.length; k++

6. 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 a 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.

7. The `twoInARow` method below is intended to return `true` if any two consecutive elements of the parameter `arr` are equal in value and return `false` otherwise.

```
public boolean twoInARow(int[] arr)
{
     /* missing loop header */
     {
         if (arr[k] == arr[k + 1])
         {
            return true;
         }
     }
     return false;
}
```

Which of the following can be used to replace `/* missing loop header */` so that the method will work as intended?

(A) `for (int k = 0; k < arr.length - 1; k++)`

(B) `for (int k = 0; k < arr.length; k++)`

(C) `for (int k = 1; k < arr.length; k++)`

(D) `for (int k = arr.length - 1; k >= 0; k--)`

(E) `for (int k = arr.length - 1; k > 0; k--)`

8. Consider the following method.

```
public static int mystery(int value)
{
     int sum = 0;
     int[] arr = {1, 4, 2, 5, 10, 3, 6, 4};

     for (int item : arr)
     {
         if (item > value)
         {
            sum += item;
         }
     }
     return sum;
}
```

What value is returned as a result of the call `mystery(4)` ?

test3v2

(A) 6

(B) 15

(C) 21

(D) 29

(E) 35