28. A boolean method find is added to the Purse class:

```java
/* Returns true if the purse has a coin that matches aCoin, * false otherwise. */
public boolean find(Coin aCoin)
{
    for (Coin c : coins)
    {
        /* code to find match */
    }
    return false;
}
```

Which is a correct replacement for /* code to find match */?

I if (c.equals(aCoin))
    return true;

II if ((c.getName()).equals(aCoin.getName()))
    return true;

III if ((c.getValue()).equals(aCoin.getValue()))
    return true;

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

29. Which of the following initializes an 8 x 10 matrix with integer values that are perfect squares? (0 is a perfect square.)

I int[][] mat = new int[8][10];

II int[][] mat = new int[8][10];
    for (int r = 0; r < mat.length; r++)
        for (int c = 0; c < mat[r].length; c++)
            mat[r][c] = r * r;

III int[][] mat = new int[8][10];
    for (int c = 0; c < mat.length; c++)
        for (int r = 0; r < mat.length; r++)
            mat[r][c] = c * c;

(A) I only
(B) II only
(C) III only
(D) I and II only
(E) I, II, and III
30. Consider the following method that will alter the matrix mat:

```java
//Precondition: mat is initialized.
public static void matStuff(int[][] mat, int row) {
    int numCols = mat[0].length;
    for (int col = 0; col < numCols; col++)
        mat[row][col] = row;
}
```

Suppose mat is originally

```
1 4 9 0  
2 7 8 6  
5 1 4 3  
```

After the method call `matStuff(mat, 2)`, matrix mat will be

(A)  
```
1 4 9 0  
2 7 8 6  
2 2 2 2  
```

(B)  
```
1 4 9 0  
2 2 2 2  
5 1 4 3  
```

(C)  
```
2 2 2 2  
2 2 2 2  
2 2 2 2  
```

(D)  
```
1 4 2 0  
2 7 2 6  
5 1 2 3  
```

(E)  
```
1 2 9 0  
2 2 8 6  
5 2 4 3  
```
Questions 34–37 use the nested for-each loop for two-dimensional arrays. This will not be tested on the AP exam.

34. The method changeNegs below should replace every occurrence of a negative integer in its matrix parameter with 0.

```java
// Precondition: mat is initialized with integers.
// Postcondition: All negative values in mat replaced with 0.
public static void changeNegs(int[][] mat)
{
    /* code */
}
```

Which is correct replacement for /* code */?

I for (int r = 0; r < mat.length; r++)
    for (int c = 0; c < mat[r].length; c++)
        if (mat[r][c] < 0)
            mat[r][c] = 0;

II for (int c = 0; c < mat[0].length; c++)
    for (int r = 0; r < mat.length; r++)
        if (mat[r][c] < 0)
            mat[r][c] = 0;

III for (int[] row : mat)
    for (int element : row)
        if (element < 0)
            element = 0;

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

"Doesn't change value in array"
35. This question is based on the Point class below:

```java
public class Point {
    private int x;
    private int y;

    // constructor
    public Point (int x, int y) {
        this.x = x;
        this.y = y;
    }

    // accessors
    public int getX() {
        return x;
    }

    public int getY() {
        return y;
    }

    // Set x and y to new_x and new_y.
    public void setPoint(int new_x, int new_y) {
        x = new_x;
        y = new_y;
    }

    // Return Point in String form.
    public String toString() {
        return "(" + x + ", " + y + ");";
    }

    // other methods not shown
}
```

The method changeNecs below takes a matrix of Point objects as parameter and
replaces every Point that has at least one negative coordinate with the Point
(0,0).

```java
/* Precondition: pointMat is initialized with Point objects.
 * Postcondition: Every point with at least one negative
 * coordinate has been changed to have both
 * coordinates equal to zero. */
public static void changeNecs (Point[][] pointMat) {
    /* code */
```
Which is a correct replacement for /* code */?

I for (int r = 0; r < pointMat.length; r++)
    for (int c = 0; c < pointMat[r].length; c++)
        if (pointMat[r][c].getx() < 0
            || pointMat[r][c].gety() < 0)
            pointMat[r][c].setPoint(0, 0);

II for (int c = 0; c < pointMat[0].length; c++)
    for (int r = 0; r < pointMat.length; r++)
        if (pointMat[r][c].getx() < 0
            || pointMat[r][c].gety() < 0)
            pointMat[r][c].setPoint(0, 0);

III for (Point[] row : pointMat)
    for (Point p : row)
        if (p.getx() < 0 || p.gety() < 0)
            p.setPoint(0, 0);

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

AB (continued)
38. The following code segment reverses the elements of \texttt{arr[first]} . . . \texttt{arr[last]}.

\begin{verbatim}
int k = first, j = last;
while (k < j)
{
    swap(arr, k, j);  //interchanges arr[k] and arr[j]
    k++;
    j--;
}
\end{verbatim}

Which of the following diagrams represents the loop invariant for the \texttt{while} loop? (Each rectangle represents a segment of array \texttt{arr}. The labels above the rectangles represent the indexes of array elements at the beginning and end of each segment.)

(A) first \hspace{1cm} k \hspace{1cm} k+1 \hspace{1cm} j-1 \hspace{1cm} j \hspace{1cm} last

(B) first \hspace{1cm} k \hspace{1cm} k+1 \hspace{1cm} j-1 \hspace{1cm} j \hspace{1cm} last

(C) first \hspace{1cm} k-1 \hspace{1cm} k \hspace{1cm} j \hspace{1cm} j+1 \hspace{1cm} last

(D) first \hspace{1cm} k-1 \hspace{1cm} k \hspace{1cm} j \hspace{1cm} j+1 \hspace{1cm} last

(E) first \hspace{1cm} k-1 \hspace{1cm} k \hspace{1cm} j-1 \hspace{1cm} j \hspace{1cm} last

39. The following algorithm sets \texttt{min} equal to the smallest value in \texttt{arr[0]} . . . \texttt{arr[n-1]}:

\begin{verbatim}
min = arr[0];
i = 1;
while (i < n)
{
    if (arr[i] < min)
    {
        min = arr[i];
i++;
    }
}
\end{verbatim}

The loop invariant for the \texttt{while} loop is

(A) \texttt{min} is smallest value in \texttt{arr[0]} . . . \texttt{arr[i]}, \( 1 \leq i \leq n \)

(B) \texttt{min} is smallest value in \texttt{arr[0]} . . . \texttt{arr[i-1]}, \( 1 \leq i \leq n-1 \)

(C) \texttt{min} is smallest value in \texttt{arr[0]} . . . \texttt{arr[i]}, \( 1 \leq i \leq n-1 \)

(D) \texttt{min} is smallest value in \texttt{arr[0]} . . . \texttt{arr[i-1]}, \( 1 < i \leq n \)

(E) \texttt{min} is smallest value in \texttt{arr[0]} . . . \texttt{arr[i-1]}, \( 1 \leq i \leq n \)
03. Consider the following code segment.

```java
ArrayList names = new ArrayList();
names.add("Tom");
names.add("Ann");
names.add("Joe");
names.add("Sue");
for (int k = 1; k <= names.size(); k++)
    System.out.print(names.get(k) + " ");
```

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

(A) IndexOutOfBoundsException message
(B) Tom Ann Joe Sue
(C) Ann Joe Sue Tom
(D) Tom Sue Joe Ann
(E) Tom Ann Joe Sue

04. Consider the following code segment.

```java
ArrayList names = new ArrayList();
names.add("Tom");
names.add("Ann");
names.add("Joe");
names.add("Sue");
for (int k = 0; k < names.size(); k++)
    System.out.print(names.get(k) + " ");
System.out.println();
for (int k = 0; k < names.size(); k++)
    System.out.print(names.get(k) + " ");
```

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

(A) Bob Bob Bob Bob Bob
(B) Tom Ann Joe Sue Bob Bob Bob Bob
(C) Bob Bob Bob Bob Tom Ann Joe Sue
(D) Bob Bob Bob Bob
(E) IndexOutOfBoundsException message
05. Consider the following code segment.

```java
ArrayList<String> names = new ArrayList();
names.add("Tom");
names.add("Ann");
names.add("Joe");
names.add("Sue");
names.add(1,"Bob");
System.out.println();
for (int k = 0; k < names.size(); k++)
    System.out.print(names.get(k) + " ");
```

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

(A) Tom Bob Joe Sue  
(B) Bob Ann Joe Sue  
(C) Tom Bob Ann Joe Sue  
(D) Bob Tom Ann Joe Sue  
(E) IndexOutOfBoundsException message

06. Consider the following code segment.

```java
ArrayList<String> names = new ArrayList();
names.add("Tom");
names.add("Ann");
names.add("Joe");
names.add("Sue");
names.add(5,"Bob");
System.out.println();
for (int k = 0; k < names.size(); k++)
    System.out.print(names.get(k) + " ");
```

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

(A) Tom Ann Joe Sue Bob  
(B) Tom Ann Joe Sue Null Bob  
(C) Tom Ann Joe Sue  
(D) Tom Ann Joe Null Bob  
(E) IndexOutOfBoundsException message
97. Consider the following code segment.

```java
ArrayList names = new ArrayList();
names.add("Tom");
names.add("Ann");
names.add("Joe");
names.add("Sue");
names.remove(1);
names.remove(2);
System.out.println();
for (int k = 0; k < names.size(); k++)
    System.out.print(names.get(k) + " ");
```

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

(A) Tom Joe
(B) Tom Sue
(C) Joe Sue
(D) Ann Sue
(E) Tom Ann

98. Consider the following code segment.

```java
ArrayList names = new ArrayList();
names.add("Tom");
names.add("Ann");
names.add("Joe");
names.add("Sue");
names.remove(1);
names.remove(3);
System.out.println();
for (int k = 0; k < names.size(); k++)
    System.out.print(names.get(k) + " ");
```

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

(A) Error message indicating that names(3) cannot be removed.
(B) IndexOutOfBoundsException message
(C) Tom Joe
(D) Ann Sue
(E) Ann Joe
Chapter 3 Questions

Arrays (A Topic 1-10, AB Topic 11-15)

01. Consider the following method.

    public static int quest0301(int n)
    {
        int temp[] = new int[n+1];
        temp[0] = 1;
        temp[1] = 1;
        for (int k = 2; k <= n; k++)
            temp[k] = temp[k-1] + temp[k-2];
        return temp[n];
    }

What value is returned by the call quest0301(10)?

(A) 21
(B) 34
(C) 45
(D) 55
(E) 89

02. Consider the following method.

    public static void quest0302(int list[])
    {
        int max = list.length-1;
        for (int k = 0; k < max; k++)
            if (list[k] < list[k+1])
            {
                int temp = list[k];
                list[k] = list[k+1];
                list[k+1] = temp;
            }
    }

Which of the following describes the result of calling method quest0302?

(A) The smallest number is located in list[max].
(B) The largest number is located in list[max].
(C) The list array is sorted in ascending order.
(D) The list array is sorted in descending order.
(E) The elements in the list array are reversed.
04. Consider the following incomplete `quest0304` method.

```java
public static int quest0304(int list[])
// precondition: Array list contains n integers.
// postcondition: quest0304 returns a₀ + a₁ + ... + aₙ₋₂ + aₙ₋₁
{
    /* missing code */
}
```

Which of the following three implementations of /* missing code */ will make method `quest0304` work as intended?

**Implementation 1**

```java
int temp = 0;
for (int k = 0; k < list.length; k++)
    temp += list[k];
return temp;
```

**Implementation 2**

```java
int temp = list[0];
int k = 0;
while (k < list.length)
{
    temp += list[k];
    k++;
}
return temp;
```

**Implementation 3**

```java
int temp = list[0];
for (int k = list.length-1; k > 0; k--)
    temp += list[k];
return temp;
```

(A) Implementation 1 only  
(B) Implementation 2 only  
(C) Implementation 3 only  
(D) Implementation 1 & 2 only  
(E) Implementation 1 & 3 only
07. Consider the following incomplete code segment.

```java
int list[] = {11,22,33,44,55,66,77,88,99};
int n = list.length-1;

/* missing code */

for (int k = 0; k < list.length; k++)
    System.out.print(list[k] + " ");
```

The output of the code segment is supposed to print the list array in reverse order.
Which of the following implementations of /* missing code */ will satisfy that requirement?

**Implementation 1**

```java
for (int k = 0; k < n; k++)
    {  
        int temp = list[k];
        list[k] = list[n-k];
        list[n-k] = temp;
    }
```

**Implementation 2**

```java
for (int k = 0; k < n/2; k++)
    {  
        int temp = list[k];
        list[k] = list[n-k];
        list[n-k] = temp;
    }
```

**Implementation 3**

```java
for (int k = 0; k < n/2; k++)
    {  
        int temp = list[k];
        list[k] = list[n];
        list[n] = temp;
    }
```

(A) Implementation 1 only  
(B) Implementation 2 only  
(C) Implementation 3 only  
(D) Implementations 2 and 3 only  
(E) Implementations 1, 2 and 3
08. Consider the following code segment.

```java
int list[] = {5, 10, 15, 20, 25, 30, 35, 40, 45, 50};
for (int k = 0; k < list.length; k++)
    list[k] = list[k] / list[0];

for (int k = 0; k < list.length; k++)
    System.out.print(list[k] + " ");
```

What will be output when the code segment executes?

(A) 1 10 15 20 25 30 35 40 45 50
(B) 1 2 3 4 5 6 7 8 9 10
(C) 0 0 0 0 0 0 0 0 0 0
(D) 1 1 1 1 1 1 1 1 1 1
(E) 5 10 15 20 25 30 35 40 45 50
Sorting

Sort the following array using:
- Bubble
- Selection
- Insertion
- Merge

Be sure to show all the stages!

<table>
<thead>
<tr>
<th>8</th>
<th>6</th>
<th>1</th>
<th>4</th>
<th>5</th>
<th>2</th>
<th>7</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>u</td>
<td>b</td>
<td>l</td>
<td>i</td>
<td>n</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>6145270</td>
<td>86145270</td>
<td>86145270</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>86145270</td>
<td>6145278</td>
<td>6145278</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6145278</td>
<td>6145278</td>
<td>16845270</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16845270</td>
<td>16845270</td>
<td>14685270</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14685270</td>
<td>14562780</td>
<td>14562780</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14562780</td>
<td>14562780</td>
<td>14562780</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Show the stages of binary search for this array if you are looking for the value 8:

<table>
<thead>
<tr>
<th>1</th>
<th>3</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>11</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lo Index</td>
<td>Hi Index</td>
<td>Mid Index</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>7</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Show the stages of binary search for this array if you are looking for the value 8.