COMPUTER SCIENCE A

SECTION I

Time—1 hour and 30 minutes

40 Questions

Directions: Determine the answer to each of the following questions or incomplete statements, using the available space for any necessary scratch work. Then decide which is the best of the choices given and fill in the corresponding circle on the answer sheet. No credit will be given for anything written in the exam booklet. Do not spend too much time on any one problem.

Notes:

- Assume that the classes listed in the Java Quick Reference have been imported where appropriate.
- Assume that declarations of variables and methods appear within the context of an enclosing class.
- Assume that method calls that are not prefixed with an object or class name and are not shown within a complete class definition appear within the context of an enclosing class.
- Unless otherwise noted in the question, assume that parameters in method calls are not `null` and that methods are called only when their preconditions are satisfied.

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

1. Consider the following code segment.

```
int x = /* some integer value */;
int y = /* some integer value */;

if (x > y && x % 2 == 0)
{
   System.out.print("armadillo");
}
else if (x + y < 30)
{
   System.out.print("koala");
}
else if (x % 2 == 0)
{
   System.out.print("platypus");
}
else
{
   System.out.print("kangaroo");
}
```

Which of the following pairs of x and y values would cause "platypus" to be printed?

(A) x = 5, y = 30

(B) x = 8, y = 5

(C) x = 8, y = 15

(D) x = 10, y = 15

(E) x = 10, y = 30

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

2. Consider the following class declaration.

```
public class SomeMethods
{
   public static String testMethod(int first, int second)
   {
      return "1";
   }

   public static String testMethod(int first, String second)
   {
      return "2";
   }

   public static String testMethod(int first)
   {
      return "3";
   }

   public static int testMethod(int value)
   {
      return 4;
   }
}
```

The following statement appears in a method in a class other than `SomeMethods`.

```
System.out.println(SomeMethods.testMethod(1234));
```

What, if anything, is printed as a result of executing the statement?

(A) 1

(B) 2

(C) 3

(D) 4

(E) Nothing is printed because the class `SomeMethods` does not compile.

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE

3. Consider the following code segment.

```
int[] arr = {4, 2, 2, 3, 1};
int num = 0;

for (int k = 0; k < arr.length; k++)
{
   num = arr[num];
}
```

After the code segment executes, what is the value of num ?

(A) 0

(B) 1

(C) 2

(D) 3

(E) 4

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

4. Consider the following code segment. Assume that `x` and `y` are properly declared and initialized integer variables.

```
int z = 0;
if (x < y)
{
   if (x > 0)
   {
      z = 10;
   }
}
else
{
   z = 20;
}
```

Under which of the following conditions is the variable `z` set to `20` ?

(A) Whenever `x` is less than `y`

(B) Whenever `x` is less than or equal to `0`

(C) Whenever `x` is less than `y` and `x` is not greater than `0`

(D) Whenever `y` is less than or equal to `x`

(E) Whenever `z` is not set to `10`

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

5. Consider the following code segment.

```
double valOne = 5.75;
double valTwo = 2.75;

int x = (int) valOne + (int) valTwo;
int y = (int) (valOne + valTwo);

System.out.println(x + "" + y);
```

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

(A) 77

(B) 78

(C) 79

(D) 98

(E) 99

---

6. Consider the following code segment.

```
int n = 0;
while (n < 10)
{
   n += 2;
   System.out.print(n + " ");
}
```

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

(A) 0 2 4 6 8

(B) 0 2 4 6 8 10

(C) 0 2 4 6 8 10 12

(D) 2 4 6 8 10

(E) 2 4 6 8 10 12

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

7. Consider the following code segment. Assume that `value` is a properly declared and initialized integer variable.

```
if (value < 7)
{
   value += value;
}
else
{
   value++;
}
```

Which of the following code segments produces the same result as the preceding code segment for all initial values of `value` ?

(A)
```
if (value != 7)
{
   value += value;
}
else
{
   value++;
}
```

(B)
```
if (value != 7)
{
   value++;
}
else
{
   value += value;
}
```

(C)
```
if (value > 7)
{
   value++;
}
else
{
   value += value;
}
```

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

(D)
```
if (value >= 7)
{
   value += value;
}
else
{
   value++;
}
```

(E)
```
if (value >= 7)
{
   value++;
}
else
{
   value += value;
}
```

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

8. Consider the following code segment.

```
for (int j = 0; j <= 3; j++)
{
   for (int k = 0; k < 3; k++)
   {
      System.out.print(j * k + " ");
   }

   System.out.println();
}
```

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

(A)
```
0 0 0
0 1 2
0 2 4
```

(B)
```
0 0 0
0 1 2
0 2 4
0 3 6
```

(C)
```
0 0 0 0
0 1 2 3
0 2 4 6
0 3 6 9
```

(D)
```
1 2 3
2 4 6
3 6 9
```

(E)
```
0 1 2 3
0 2 4 6
0 3 6 9
```

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

9. The `Toaster` class is a subclass of the `Appliance` class. The `Toaster` class has an instance variable `maxSlices`, which is the maximum number of slices of toast that the toaster can make at the same time. Which of the following is a correct partial definition of the `Toaster` class?

(A)
```
public class Appliance
{
   private Toaster maxSlices;
   // There may be instance variables, constructors, and methods not shown.
}
```

(B)
```
public class Appliance extends Toaster
{
   private int maxSlices;
   // There may be instance variables, constructors, and methods not shown.
}
```

(C)
```
public class Toaster
{
   private int maxSlices;
   // There may be instance variables, constructors, and methods not shown.
}
```

(D)
```
public class Toaster extends Appliance
{
   private int maxSlices;
   // There may be instance variables, constructors, and methods not shown.
}
```

(E)
```
public class Toaster extends Appliance
{
   private Toaster maxSlices;
   // There may be instance variables, constructors, and methods not shown.
}
```

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

10. Assume that x and y are boolean variables and have been properly initialized.

```
(x && y) || y
```

Which of the following always evaluates to the same value as the preceding expression?

(A) `x`

(B) `y`

(C) `x && y`

(D) `x || y`

(E) `x != y`

---

11. Consider the following method, which is intended to print the `ArrayList names` in reverse order.

```
public static void printInReverseOrder(ArrayList<String> names)
{
   for ( /* missing control expressions */ )
   {
      System.out.println(names.get(k));
   }
}
```

Which of the following can be used to replace */* missing control expressions */* so that `printInReverseOrder` will work as intended?

(A) `int k = 0; k < names.size(); k++`

(B) `int k = 0; k < names.size() - 1; k++`

(C) `int k = names.size(); k >= 0; k--`

(D) `int k = names.size() - 1; k >= 0; k--`

(E) `int k = names.size() - 1; k > 0; k--`

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

12. Consider the following method.

```
public static int calc(int n)
{
   if (n == 0)
   {
      return 0;
   }
   else
   {
      return calc(n - 1) + n;
   }
}
```

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

(A) 0

(B) 1

(C) 3

(D) 6

(E) 10

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

13. Consider the following code segment.

```
int[][] rectangle = new int[4][5];
/* some statements not shown */
x = rectangle[0];
```

Assume that `x` has been previously declared such that the code segment compiles without error. Which of the following best describes the contents of `x` after the code segment has been executed?

(A) An int array containing the entire first row of rectangle

(B) An int array containing the entire first column of rectangle

(C) An int array containing the full contents of rectangle

(D) An int value representing the length of the first row of rectangle

(E) The int value in the first row and first column of rectangle

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

14. Consider the following code segment.

```
int t = 0;
for (int v = 1; v < 20; /* missing increment */ )
{
   t = t + v;
}
```

Which of the following can be used to replace `/* missing increment */` so that the variable `t` has the value `13` as a result of the execution of the code segment?

(A) `v = v * 0`

(B) `v = v * 1`

(C) `v = v * 2`

(D) `v = v * 3`

(E) `v = v * 4`

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

15. Consider a method to determine whether two arrays have the same length and same values stored in the same order.

```
public static boolean haveSameContents(int[] arr1, int[] arr2)
{  /* implementation not shown */  }
```

Consider the following implementations of `haveSameContents`.

I.
```
return (arr1.length == arr2.length) && (arr1 == arr2);
```

II.
```
if (arr1.length != arr2.length)
{
   return false;
}
for (int k = 0; k < arr1.length; k++)
{
   return arr1[k] == arr2[k];
}
return true;
```

III.
```
if (arr1.length == arr2.length)
{
   for (int k = 0; k < arr1.length; k++)
   {
      if (arr1[k] != arr2[k])
      {
         return false;
      }
   }
   return true;
}
return false;
```

Which of the implementations can be used so that `haveSameContents` will work as intended?

(A) I only

(B) II only

(C) III only

(D) I and III

(E) II and III

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

16. Consider the following code segment.

```
int x = 102;
int y = 5;
int c = 0;
while (x - y >= 0)
{
   x = x - y;
   c++;
}
```

What is the value of c after the code segment has been executed?

(A) 0

(B) 2

(C) 20

(D) 97

(E) The value of e cannot be determined because the loop does not terminate.

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

17. Consider the following class definition.

```
public class Container
{
   private static int maxInitialOrder = 0;
   private int numOfItems;

   public Container(int n)
   {
      numOfItems = n;
      if (n > maxInitialOrder)
      {
         maxInitialOrder = n;
      }
   }

   public void addItems(int more)
   {
      numOfItems += more;
   }

   public double calcShippingCost()
   {
      if (numOfItems == maxInitialOrder)
      {
         return 0.0;
      }
      else
      {
         return numOfItems * 0.5;
      }
   }
}
```

The following code segment appears in a class other than `Container`. Assume that no other `Container` objects have been created before the code segment is executed.

```
Container c1 = new Container(10);
Container c2 = new Container(6);
c1.addItems(2);
c2.addItems(4);
double cost = c2.calcShippingCost();
```

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

What value will be assigned to cost as a result of executing the code segment?

(A) 0.0

(B) 2.0

(C) 3.0

(D) 5.0

(E) 6.0

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

18. Consider the following methods.

```
public static int alpha(int x)
{
   x = x * 2;
   System.out.print(x);
   return x + 1;
}

public static int beta(int y)
{
   System.out.print(y);
   return y + 6;
}
```

The following code segment appears in a method in the same class as `alpha` and `beta`.

```
int w = 1;
if (alpha(w) > 0 || beta(w) < 0)
{
   System.out.print(alpha(w));
}
else
{
   System.out.print(beta(w));
}
```

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

(A) 223

(B) 245

(C) 2117

(D) 2123

(E) 2245

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

19. Consider the method `replaceStr`, which is intended to return a string that is identical to `orig` except that all occurrences of string `a` are replaced by string `b`.

```
/** Precondition: a.length() == 1; b.length() == 1 */
public static String replaceStr(String orig, String a, String b)
{
   int position = orig.indexOf(a);
   while (position != -1)
   {
      orig = orig.substring(0, position) + b +
             orig.substring(position + 1);
      position = orig.indexOf(a);
   }
   return orig;
}
```

Which of the following statements can be used to show that `replaceStr` does not always work as intended?

(A) `String str = replaceStr("", "C", "D");` `// str should be set to ""`

(B) `String str = replaceStr("AB", "C", "D");` `// str should be set to "AB"`

(C) `String str = replaceStr("C", "C", "D");` `// str should be set to "D"`

(D) `String str = replaceStr("CCC", "C", "D");` `// str should be set to "DDD"`

(E) `String str = replaceStr("CCD", "C", "C");` `// str should be set to "CCD"`

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

20. Which of the following Java expressions evaluates to `true` if the `int` variable `number` is positive or is between `-10` and `-5`, inclusive, and evaluates to `false` otherwise?

(A) `(-10 <= number <= -5) || number > 0`

(B) `(-10 <= number || number <= -5) || number > 0`

(C) `(-10 <= number || number <= -5) && number > 0`

(D) `(-10 <= number && number <= -5) || number > 0`

(E) `(-10 <= number && number <= -5) && number > 0`

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

21. Consider the following method.

```
public static String doSomething(int current)
{
   if (current <= 0)
   {
      return "";
   }
   else
   {
      return doSomething(current - 2) + current;
   }
}
```

What value is returned as a result of the call `doSomething(10)` ?

(A) `"108642"`

(B) `"246810"`

(C) `"0246810"`

(D) `"1086420"`

(E) `"12345678910"`

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

22. Consider the following method, which is intended to print the contents of a two-dimensional array of integers in row-major order.

```
public static void print2DArray(int[][] table)
{
   for (int a = 0; a < table.length; a++)
   {
      for (int b = 0; b < table[0].length; b++)
      {
         System.out.print(table[b][a] + " ");
      }
      System.out.println();
   }
}
```

The method does not always work as intended. In some cases it throws an `ArrayIndexOutOfBoundsException`. Which of the following identifies the error in the method?

(A) Variables `a` and `b` are incorrectly initialized.

(B) The loop conditions use the incorrect relational operator.

(C) The outer and inner for loops contain different Boolean expressions.

(D) The loop control variable increments `a++` and `b++` are incorrect.

(E) The row and column indices in the print method call are reversed.

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

23. Consider the following code segment.

```
int[] items = {10, 20, 30, 40, 50, 60};
int sum = 0;

for (int k = items.length; k > items.length / 2; k--)
{
   sum += items[k - 1];
   items[k - 1] = items[k - 2];
}

System.out.println(sum);
```

What, if anything, will be printed as a result of executing the code segment?

(A) 90

(B) 110

(C) 150

(D) 180

(E) Nothing is printed because an `ArrayIndexOutOfBoundsException` is thrown.

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

24. The following method prints the value of `start` and then prints every other value up to, and possibly including, `end`. For example, the method call `printRun(1, 7)` prints the values `1`, `3`, `5`, and `7`.

```
/** Precondition: start < end */
public static void printRun(int start, int end)
{
   for (int k = start; k <= end; k += 2)
   {
      System.out.println(k);
   }
}
```

Consider a modification to the method that changes the Boolean expression in the for loop header so that the for loop header becomes the following.

```
for (int k = start; k < end; k += 2)
```

Which of the following best explains how the output of the modified method differs from the output of the original method?

(A) The modified method always prints fewer values than the original method.

(B) The modified method prints fewer values than the original method only when start is even.

(C) The modified method prints fewer values than the original method only when start and end are both even or when start and end are both odd.

(D) The modified method prints fewer values than the original method only when `start` is even and `end` is odd or when `start` is odd and `end` is even.

(E) The modified method never prints fewer values than the original method.

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

25. Consider the following method.

```
/** Precondition: lim >= 0 */
public static void printletters (int lim)
{
   for (int m = 0; m < lim; m++)
   {
      for (int n = 0; n < lim; n++)
      {
         System.out.print("Z");
      }
   }
}
```

Assume that `int k` has been initialized with a value that meets the precondition of the method. Which of the following expressions represents the number of times `"Z"` will be printed as a result of executing the call `printLetters(k)` ?

(A) `0`

(B) `k`

(C) `2 * k`

(D) `(k - 1) * (k - 1)`

(E) `k * k`

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

26. The following incomplete method is intended to simulate a spinner in a game by returning `"red"`, `"blue"`, or `"green"` with equal probability.

```
public static String spin()
{
   double randNum = Math.random();
   /* missing code */
}
```

The following code segments have been proposed to replace `/* missing code */`.

I.
```
int num = (int)(randNum * 3);
if (num == 0)
{
   return "red";
}
else if (num == 1)
{
   return "blue";
}
else
{
   return "green";
}
```

II.
```
if (randNum < 1 / 3.0)
{
   return "red";
}
else if (randNum < 2 / 3.0)
{
   return "blue";
}
else
{
   return "green";
}
```

III.
```
String result;
if (randNum < 1 / 3.0)
{
   result = "red";
}
if (randNum < 2 / 3.0)
{
   result = "blue";
}
else
{
   result = "green";
}
return result;
```

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

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

(A) I only

(B) I and II only

(C) I and III only

(D) II and III only

(E) I, II, and III

GO ON TO THE NEXT PAGE.

Unauthorized copying or reuse of any part of this page is illegal.

Questions 27–28 refer to the following information.

Consider the following correct implementation of the selection sort algorithm. The method sorts an `ArrayList` of integers into ascending order.

```
public static void selectionSort(ArrayList<Integer> elements)
{
   for (int j = 0; j < elements.size() - 1; j++)      // Line 1
   {
      int index = j;
      for (int k = j + 1; k < elements.size(); k++)   // Line 2
      {
         if (elements.get(k) < elements.get(index))   // Line 3
         {
            index = k;
         }
      }
      int temp = elements.get(j);
      elements.set(j, elements.get(index));
      elements.set(index, temp);

      /* End of outer loop */
   }
}
```

27. Assume that `selectionSort` is called with an `ArrayList` parameter that has been initialized with the following `Integer` objects.

    `[2, 3, 1, 5, 6, 1]`

    What will the contents of `elements` be after three passes of the outer loop (i.e., when `j == 2` at the point indicated by `/* End of outer loop */`) ?

    (A) `[1, 1, 2, 3, 5, 6]`

    (B) `[1, 1, 2, 3, 6, 5]`

    (C) `[1, 1, 2, 5, 6, 3]`

    (D) `[1, 2, 3, 5, 6, 1]`

    (E) `[1, 3, 2, 5, 6, 1]`

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

28. The method can be revised to sort the Arraylist into descending order by changing some or all occurrences of the < operator to the > operator. The lines in the `selectionSort` method marked Line 1, Line 2, and Line 3 are lines where the < operator is used. In which lines should the < operator be replaced with the > operator so that `selectionSort` sorts the `ArrayList` into descending order?

(A) Lines 1 and 2 only

(B) Lines 1 and 3 only

(C) Lines 1, 2, and 3

(D) Lines 2 and 3 only

(E) Line 3 only

GO ON TO THE NEXT PAGE.

Unauthorized copying or reuse of any part of this page is illegal.

29. Consider the following class declarations.

```
public class Alpha
{
   private int thing;

   public Alpha()
   {
      thing = 0;
   }

   public Alpha(int val)
   {
      thing = val;
   }
}

public class Beta extends Alpha
{
   public Beta()
   {
      /* missing implementation */
   }
}
```

Which of the following can be used to replace `/* missing implementation */` so that the constructor for `Beta` initializes the value of `thing` to `7` ?

(A) `thing = 7;`

(B) `Alpha.thing = 7;`

(C) `super(7);`

(D) `super.thing = 7;`

(E) `super.Alpha(7);`

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

30. Consider the following code segment.

```
int r = 23;
int t = 10;
double a = r % t;
double b = r / t;
System.out.println(a + " " + b);
```

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

(A) `2.0 3.0`

(B) `2.3 3.0`

(C) `3.0 2.0`

(D) `3.0 2.3`

(E) Nothing is printed. A compile-time error occurs because an `int` value cannot be assigned to a `double`.

GO ON TO THE NEXT PAGE.

Unauthorized copying or reuse of any part of this page is illegal.

31. Consider the following code segment.

```
for (int j = 0; j < 4; j++)
{
   for ( /* missing code */ )
   {
      System.out.print("(" + j + "," + k + ")" + " ");
   }
   System.out.println();
}
```

The code segment is intended to print the following output.

```
(0,1) (0,2) (0,3)
(1,2) (1,3)
(2,3)
```

Which of the following can replace `/* missing code */` so that the code segment works as intended?

(A) `int k = 0; k < j; k++`

(B) `int k = j; k < j; k++`

(C) `int k = j; k < 4; k++`

(D) `int k = j + 1; k < j; k++`

(E) `int k = j + 1; k < 4; k++`

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

32. Consider the following code segment, which is intended to print `true` if all elements of the integer array `arr` are equal in value and `false` otherwise. Assume `arr` is properly declared and initialized and has at least two elements.

```
boolean result = true;

for (int k = 0; k < arr.length; k++)
{
   if (arr[k] != arr[k + 1])
   {
      result = false;
   }
}
System.out.println(result);
```

Which of the following best explains why the code segment does not work as intended?

(A) The code segment accesses a `null` array element, causing a `NullPointerException`.

(B) The code segment does not compare all pairs of array elements.

(C) The code segment incorrectly initializes the variable result.

(D) The code segment sets the variable result to false under the wrong conditions.

(E) The code segment encounters an invalid array index, causing an `ArrayIndexOutOfBoundsException`.

GO ON TO THE NEXT PAGE.

Unauthorized copying or reuse of any part of this page is illegal.

33. The following method is intended to return a string that is formed by exchanging the first and last characters of its string parameter. For example, the call `switchLetters("wagon")` should return the string `"nagow"`.

```
/** Precondition: str.length() >= 2 */
public static String switchletters (String str)
{
   int len = str.length();
   String result;
   /* missing code */
   return result;
}
```

The following code segments have been proposed as replacements for `/* missing code */`.

I.
```
result = str.substring(len - 1);
result += str.substring(1, len - 1);
result += str.substring(0, 1);
```

II.
```
result = str.substring(len - 1);
result += str.substring(1);
result += str.substring(0, 1);
```

III.
```
result = str.substring(0, len - 1);
result += str.substring(0, 1);
result = str.substring(len - 1) + result.substring(1);
```

Which of the code segments can be used to replace `/* missing code */` so that the `switchLetters` method works as intended?

(A) I only

(B) I and II only

(C) I and III only

(D) II and III only

(E) I, II, and III

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

34. Consider the following incomplete method, which is intended to reverse the elements in an array.

```
public static void reverse(int[] values)
{
   for (int k = 0; /* condition */; k++)
   {
      int temp = values[k];
      int swapIndex = /* expression */;
      values[k] = values[swapIndex];
      values[swapIndex] = temp;
   }
}
```

For example, if the method is called with the following array

5	8	3	7	4	6	2

,

then, upon completion, the array should contain the following.

2	6	4	7	3	8	5

Which of the following can be used to replace `/* condition */` and `/* expression */` so that `reverse` will work as intended?

	`/* condition */`	*`/* expression */`*
(A)	`k < values.length`	`values.length - k - 1`
(B)	`k < values.length`	`values.length - k`
(C)	`k < values.length / 2`	`values.length - k - 1`
(D)	`k < values.length / 2`	`values.length - k`
(E)	`k < values.length / 2`	`values.length - k + 1`

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

35. Consider the following class definition.

```
public class ArrayTester
{
   /** Returns true if r is special. */
   public static boolean isSpecial(int[] r)
   {  /* implementation not shown */  }

   /** Returns true if num is incredible. */
   public static boolean isIncredible(int num)
   {  /* implementation not shown */  }

   public static void check(int[][] data)
   {
      boolean result1 = false;
      boolean result2 = true;

      for (int[] d : data)
      {
         if (isSpecial(d))
         {
            result1 = true;
         }
         for (int n : d)
         {
            result2 = result2 && isIncredible(n);
         }
      }

      System.out.println(result1 + " " + result2);
   }
}
```

Assume that `someArray` is a properly declared and initialized two-dimensional array of integers. Which of the following best describes the conditions under which the method call `ArrayTester.check(someArray)` prints the output `true true` ?

(A) There are no special rows in `someArray`, and at least one value in `someArray` is incredible.

(B) All rows in `someArray` are special, and at least one value in `someArray` is incredible.

(C) All rows in `someArray` are special, and all values in `someArray` are incredible.

(D) At least one row in `someArray` is special, and at least one value in `someArray` is incredible.

(E) At least one row in `someArray` is special, and all values in `someArray` are incredible.

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

36. Consider the following class `BookInfo`.

```
public class BookInfo
{
   private String author;
   private String bookName;

   /** Creates a BookInfo object where name is the author's name and
    *  title is the title of the book.
    */
   public BookInfo(String name, String title)
   {
      author = name;
      bookName = title;
   }
}
```

Which of the following would be an appropriate method header for a `BookInfo` method that would return the title of the book when called from a method in a class other than `BookInfo` ?

(A) `public void getBookName(String title)`

(B) `public String getBookName()`

(C) `private String getBookName(String title)`

(D) `private String getBookName()`

(E) `private void getBookName(String title)`

GO ON TO THE NEXT PAGE.

Unauthorized copying or reuse of any part of this page is illegal.

37. A rectangular playing board is to be set up so that in the even-numbered rows, there is an `x` in each odd-numbered column, and in the odd-numbered rows, there is an `x` in each even-numbered column.

The following table shows how a 4×8 playing board would be initialized.

	0	1	2	3	4	5	6	7
0		x		x		x		x
1	x		x		x		x	
2		x		x		x		x
3	x		x		x		x	

Consider the following incomplete code segment. Assume that `board` has been declared appropriately and has been initialized with strings containing a single blank character.

```
for (int row = 0; row < board.length; row++)
{
   for (int col = /* expr */; col < board[0].length; col = col + 2)
   {
      board[row][col] = "x";
   }
}
```

Which of the following can be used to replace `/* expr */` so that the code segment will work correctly?

(A) `row / 2`

(B) `(row / 2) + 1`

(C) `row % 2`

(D) `(row % 2) + 1`

(E) `(row + 1) % 2`

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

38. Consider the following method.

```
public static String processWords(String[] words, int index)
{
   String result;
   if (index >= words.length)
   {
      result = "";
   }
   else
   {
      result = processWords(words, index + 1) + words[index];
   }

   return result;
}
```

The following code segment appears in a method in the same class as processwords.

```
String[] things = {"Bear", "Apple", "Gorilla", "House", "Car"};
System.out.println(processWords(things, 2));
```

What will be printed as a result of executing the code segment?

(A) GorillaAppleBear

(B) GorillaHouseCar

(C) CarHouseGorilla

(D) BearAppleGorilla

(E) AppleGorillaHouseCar

GO ON TO THE NEXT PAGE.

Unauthorized copying or reuse of any part of this page is illegal.

39. Consider the following class definition.

```
public class Item
{
   private int price;

   /** Postcondition: price is initialized to p. */
   public Item(int p)
   {
      /* implementation not shown */
   }

   public int getPrice()
   {
      return price;
   }

   /* missing method */
}
```

The following code segment appears in a class other than Item. The code segment is intended to change the value of the `price` instance variable of the `item1` object from `5` to `6`.

```
Item item1 = new Item(5);
item1.setPrice(6);
```

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

Which of the following methods could replace /* *missing method* */ so that the code segment works as intended?

(A)
```
public void setPrice(int p)
{
   price = p;
}
```

(B)
```
public void setPrice(int p)
{
   price = getPrice();
}
```

(C)
```
public void setPrice(int p)
{
   p = getPrice();
}
```

(D)
```
public int setPrice(int p)
{
   p = price;
   return price;
}
```

(E)
```
public int setPrice(int p)
{
   p = getPrice();
   return p;
}
```


GO ON TO THE NEXT PAGE.

Unauthorized copying or reuse of any part of this page is illegal.

40. Consider the following class definition.

```
public class WordPlay
{
   private String word;

   public WordPlay(String w)
   {  word = w;  }

   public void update(String first)
   {  word = first + word;  }

   public void update(String first, String second)
   {  word = word.indexOf(first) + second;  }

   public String getWord()
   {  return word;  }
}
```

Assume that the following code segment appears in a class other than `WordPlay`.

```
WordPlay wp = new WordPlay("bandana");
wp.update("d", "tie");
wp.update("scarf");
System.out.println(wp.getWord());
```

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

(A) `scarf3tie`

(B) `scarfdanatie`

(C) `tie3scarf`

(D) `tiedanascarf`

(E) Nothing is printed. The `WordPlay` class does not compile because it contains two `update` methods.

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

COMPUTER SCIENCE A

SECTION II

Time—1 hour and 30 minutes

4 Questions

Directions: SHOW ALL YOUR WORK. REMEMBER THAT PROGRAM SEGMENTS ARE TO BE WRITTEN IN JAVA. You may plan your answers in this orange booklet, but no credit will be given for anything written in this booklet. **You will only earn credit for what you write in the separate Free Response booklet.**

Notes:

- Assume that the classes listed in the Java Quick Reference have been imported where appropriate.
- Unless otherwise noted in the question, assume that parameters in method calls are not `null` and that methods are called only when their preconditions are satisfied.
- In writing solutions for each question, you may use any of the accessible methods that are listed in classes defined in that question. Writing significant amounts of code that can be replaced by a call to one of these methods will not receive full credit.

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

NO TEST MATERIAL ON THIS PAGE

GO ON TO THE NEXT PAGE.

1. This question involves a game in which a player tries to hit a target by throwing a ball. You will write two `static` methods of the following `BallGame` class.

```
public class BallGame
{
    /** Simulates a player throwing the ball in the game and returns a score, as described
     *  in part (a)
     */
    public static int ballThrow()
    {  /* to be implemented in part (a) */  }

    /** Simulates a player throwing the ball numThrows times and returns the average of all throws
     *  with scores greater than minScore, as described in part (b)
     *  Precondition: numThrows > 0
     */
    public static double averageThrow(int numThrows, int minScore)
    {  /* to be implemented in part (b) */  }

    // There may be instance variables, constructors, and methods that are not shown.
}
```

(a) Write the `ballThrow` method, which simulates a player throwing a ball one time. The method returns a score of `10`, `20`, `30`, `40`, or `50`. The score is randomly generated, with each score having an equal chance of being returned.

Complete method `ballThrow`.

```
/** Simulates a player throwing the ball in the game and returns a score, as described in part (a) */
public static int ballThrow()
```

Begin your response at the top of a new page in the Free Response booklet and fill in the appropriate circle indicating the question number.
If there are multiple parts to this question, write the part letter with your response.

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

(b) Write the `averageThrow` method, which simulates a player throwing the ball `numThrows` times and returns the average of all throws with scores greater than `minScore`.

If none of the throws have a score greater than `minScore`, then the `averageThrow` method returns `0.0`.

The following table represents four calls to `averageThrow` and the possible results.

Call to `averageThrow`	Possible Results Generated by `ballThrow`	Value Returned by `averageThrow`	Explanation
`BallGame.averageThrow(3, 10)`	`30, 40, 50`	`40.0`	The average of `30, 40,` and `50`
`BallGame.averageThrow(3, 20)`	`50, 10, 40`	`45.0`	The average of `50` and `40`
`BallGame.averageThrow(4, 0)`	`10, 20, 10, 10`	`12.5`	The average of `10, 20, 10,` and `10`
`BallGame.averageThrow(4, 30)`	`20, 20, 30, 10`	`0.0`	Since none of the throws have a score greater than `30`, the `averageThrow` method returns `0.0`.

Complete method `averageThrow`. Assume that `ballThrow` works as specified, regardless of what you wrote in part (a). You must use `ballThrow` appropriately to receive full credit.

```
/** Simulates a player throwing the ball numThrows times and returns the average of all throws
 *  with scores greater than minScore, as described in part (b)
 *  Precondition: numThrows > 0
 */
public static double averageThrow(int numThrows, int minScore)
```

Begin your response at the top of a new page in the Free Response booklet and fill in the appropriate circle indicating the question number.

If there are multiple parts to this question, write the part letter with your response.

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

2. The `SecretWord` class is used to store and transform a string.

```
public class SecretWord
{
    /** The secret word */
    private String original;

    /** Constructs a SecretWord object with the string word
     *  Precondition: word is not an empty String.
     */
    public SecretWord(String word)
    {  /* implementation not shown */  }

    /** Returns original */
    public String getOriginal()
    {  /* implementation not shown */  }

    /** Returns a scrambled version of original, which has the same length as original
     *  Postcondition: original is unchanged.
     */
    public String transformWord()
    {  /* implementation not shown */  }
}
```

You will write a class `TopSecretWord`, which is a subclass of `SecretWord`. The `TopSecretWord` class further transforms the scrambled version of the original string based on the following rules, in a way that may change the length of the string.

- If the length of the string is even, replace the first half of the scrambled string with the string `"***"`.
- If the length of the string is odd, replace the second half of the scrambled string with the string `"***"`. Note that if a string has an odd number of letters, the second half of the string contains the extra letter.

The `TopSecretWord` class contains an additional method, `checkLength`, which returns `true` if the length of the result of the `TopSecretWord` transformation is greater than 5 and returns `false` otherwise.

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

The following table contains a sample code execution sequence and the corresponding results. The code execution sequence appears in a class other than `SecretWord` or `TopSecretWord`.

Statement	Return Value (blank if none)	Class Specification
`SecretWord w1 = new SecretWord("Programs");`		`w1` is a `SecretWord` created with the string `"Programs"`.
`w1.transformWord();`	`"oPmrarsg"`	This scrambled string is one possible return value. The return value is always a string containing the scrambled letters of the original string.
`w1.getOriginal();`	`"Programs"`	The original string is returned.
`TopSecretWord w2 = new TopSecretWord("pixelated");`		`w2` is a `TopSecretWord` created with the string `"pixelated"`.
`w2.transformWord();`	`"xtai***"`	If the `SecretWord` transformed string is `"xtaipdele"`, the return value is a string containing the first half of `"xtaipdele"` followed by `"***"`, because the original word has an odd length.
`w2.transformWord();`	`"tale***"`	If the `SecretWord` transformed string is `"talexipde"`, the return value is a string containing the first half of `"talexipde"` followed by `"***"`, because the original word has an odd length.
`w2.getOriginal();`	`"pixelated"`	The original string is returned.
`TopSecretWord w3 = new TopSecretWord("bird");`		`w3` is a `TopSecretWord` created with the string `"bird"`.
`w3.transformWord();`	`"***di"`	If the `SecretWord` transformed string is `"rbdi"`, the return value is a string containing `"***"` followed by the second half of `"rbdi"`, because the original word has an even length.
`w2.checkLength();`	`true`	The length of the transformed word (7) is greater than 5.
`w3.checkLength();`	`false`	The length of the transformed word (5) is not greater than 5.

Write the complete `TopSecretWord` class. Your implementation must meet all specifications and conform to the examples.

Begin your response at the top of a new page in the Free Response booklet and fill in the appropriate circle indicating the question number.
If there are multiple parts to this question, write the part letter with your response.

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

3. This question involves reasoning about an array of integers in which there are no duplicate values. An *inversion* is a pair of values in an array in which a larger value appears before a smaller value. In array `arr`, `(arr[j], arr[k])` is an inversion when `arr[j] > arr[k]` and `j < k`. For example, in the array `{-3, 4, 2, 7, 1, 6}` there are five inversion pairs: (4, 2), (4, 1), (2, 1), (7, 1), and (7, 6).

You will write two `static` methods of the following `ArrayChecker` class.

```
public class ArrayChecker
{
    /** Returns a list of integers representing the inversion pairs from the array numbers,
     *  as described in part (a)
     *  Precondition: numbers.length >= 2 and numbers contains no duplicate values.
     */
    public static ArrayList<Integer> allInversions(int[] numbers)
    {  /* to be implemented in part (a) */  }

    /** Returns a value from the array numbers that appears in the greatest number of
     *  inversion pairs, as described in part (b)
     *  Precondition: There is at least one inversion pair in numbers.
     */
    public static int valueWithMostInversions(int[] numbers)
    {  /* to be implemented in part (b) */  }

    /** Returns the number of occurrences of value in nums */
    public static int countOccur(ArrayList<Integer> nums, int value)
    {  /* implementation not shown */  }
}
```

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

(a) Write the `ArrayChecker` method `allInversions`, which returns a list of integers representing the inversion pairs from the array `numbers`. The two values in each inversion pair must be adjacent entries in the returned list, with the larger value appearing before the smaller value. The inversion pairs may appear in any order in the returned list, and each inversion pair appears exactly once in the returned list.

For example, assume that the following code segment is executed.

```
int[] sequence = {-3, 4, 2, 7, 1, 6};
ArrayList<Integer> inversions = ArrayChecker.allInversions(sequence);
```

The five inversion pairs identified from the data in the array sequence may appear in any order in the list inversions. The following shows two possible orderings of the five inversion pairs and the corresponding contents of the list `inversions`.

	Possible Ordering	`inversions` List to Be Returned
Ordering 1	(4, 2), (4, 1), (2, 1), (7, 1), (7, 6)	`[4, 2, 4, 1, 2, 1, 7, 1, 7, 6]`
Ordering 2	(7, 6), (7, 1), (2, 1), (4, 1), (4, 2)	`[7, 6, 7, 1, 2, 1, 4, 1, 4, 2]`

Complete method `allInversions`.

```
/** Returns a list of integers representing the inversion pairs from the array numbers,
 *  as described in part (a)
 *  Precondition: numbers.length >= 2 and numbers contains no duplicate values.
 */
public static ArrayList<Integer> allInversions(int[] numbers)
```

Begin your response at the top of a new page in the Free Response booklet and fill in the appropriate circle indicating the question number.

If there are multiple parts to this question, write the part letter with your response.

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

(b) Write the `ArrayChecker` method `valueWithMostInversions`, which returns a value from the array `numbers` that appears in the greatest number of inversion pairs. If more than one value appears in the maximum number of inversion pairs, any one of those values may be returned.

For example, assume that the following code segment is executed.

```
int[] sequence = {-3, 4, 2, 7, 1, 6};
ArrayList<Integer> inversions = ArrayChecker.allInversions(sequence);
```

After executing the code segment, inversions contains
`[4, 2, 4, 1, 2, 1, 7, 1, 7, 6]`.

The method call `ArrayChecker.valueWithMostInversions(sequence)` would return the value `1`, because `1` appears the greatest number of times in `inversions`.

A helper method `countOccur(ArrayList<Integer> nums, int value)` has been provided for you. The method returns the number of times `value` occurs in `nums`.

Complete method `valueWithMostInversions`. Assume that `allInversions` works correctly regardless of what you wrote for part (a). You must use `allInversions` and `countOccur` appropriately to receive full credit.

```
/** Returns a value from the array numbers that appears in the greatest number of inversion pairs,
 *  as described in part (b)
 *  Precondition: There is at least one inversion pair in numbers.
 */
public static int valueWithMostInversions(int[] numbers)
```

Begin your response at the top of a new page in the Free Response booklet and fill in the appropriate circle indicating the question number.
If there are multiple parts to this question, write the part letter with your response.

Class information for this question

```
public class ArrayChecker
public static ArrayList<Integer> allInversions(int[] numbers)
public static int valueWithMostInversions(int[] numbers)
public static int countOccur(ArrayList<Integer> nums, int value)
```

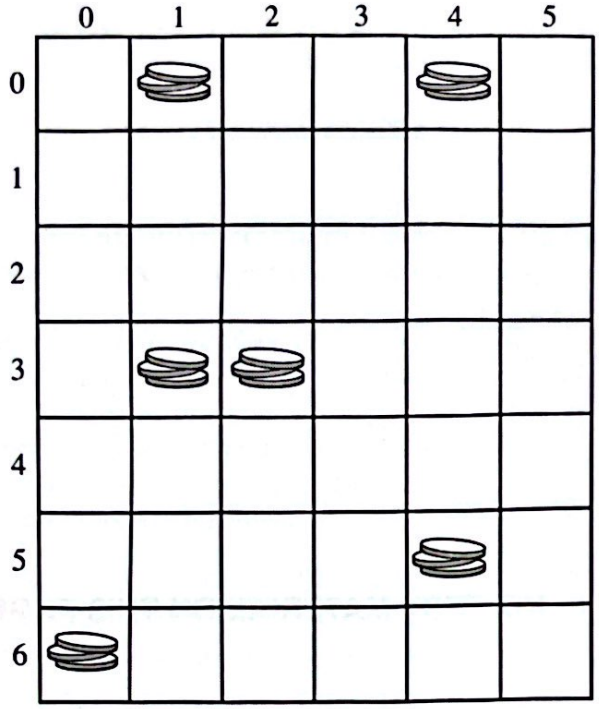
Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

NO TEST MATERIAL ON THIS PAGE

GO ON TO THE NEXT PAGE.

4. A rectangular two-dimensional (2D) array named `map` represents an area in which treasures are hidden, and `map[r][c]` represents the element at row `r` and column `c`. For example, in the `map` array shown below, elements containing treasures have a picture of a stack of coins, each of which represents an object of type `Treasure`.



Treasure objects are defined by the partial class shown below.

```
public class Treasure
{
    /** Creates a Treasure object with a positive random number of gold coins */
    public Treasure()
    {  /* implementation not shown */  }

    /** Returns the number of gold coins in this Treasure object */
    public int getGold()
    {  /* implementation not shown */  }

    // There may be instance variables, constructors, and methods that are not shown.
}
```

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

The Location class shown below represents a row and column coordinate.

```
public class Location
{
    /** Creates a Location object with the specified row and column */
    public Location(int row, int col)
    {  /* implementation not shown */  }

    /** Returns the row index of this location */
    public int getRow()
    {  /* implementation not shown */  }

    /** Returns the column index of this location */
    public int getCol()
    {  /* implementation not shown */  }

    // There may be instance variables, constructors, and methods that are not shown.
}
```

A treasure map is represented by the following `TreasureMap` class. You will write the constructor and the method `totalGold` in the `TreasureMap` class.

```
public class TreasureMap
{
    /** The map */
    private Treasure[][] map;

    /** Constructs a treasure map, as described in part (a)
     *  Precondition: r > 0, c > 0, and the size of locs is at least 1 and less than r * c.
     *        All locations in locs are valid on the map, and there are no duplicates.
     *  Postcondition: map contains r rows and c columns.
     *        map contains Treasure objects at each location in locs.
     *        All other map elements are null.
     */
    public TreasureMap(int r, int c, ArrayList<Location> locs)
    {  /* to be implemented in part (a) */  }

    /** Returns the total number of gold coins found in all map locations within the
     *  rectangular region defined by start and end, as described in part (b)
     *  Precondition: start and end are valid locations in map.
     *    The row index of start is less than or equal to the row index of end.
     *    The column index of start is less than or equal to the column index of end.
     */
    public int totalGold(Location start, Location end)
    {  /* to be implemented in part (b) */  }
}
```

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

(a) Write the constructor for the `TreasureMap` class. The constructor must initialize the instance variable `map`. The instance variable `map` must have the given number of rows and columns and contain `Treasure` objects for only the locations in the `ArrayList locs`. All locations in `locs` are valid and within the bounds of `map`, and there are no duplicates.

Complete the `TreasureMap` constructor.

```
/** Constructs a treasure map, as described in part (a)
 *  Precondition: r > 0, c > 0, and the size of locs is at least 1 and less than r * c.
 *       All locations in locs are valid on the map, and there are no duplicates.
 *  Postcondition: map contains r rows and c columns.
 *       map contains Treasure objects at each location in locs.
 *       All other map elements are null.
 */
public TreasureMap(int r, int c, ArrayList<Location> locs)
```

Begin your response at the top of a new page in the Free Response booklet and fill in the appropriate circle indicating the question number.
If there are multiple parts to this question, write the part letter with your response.

Class information for this question

```
public class Treasure

public Treasure()
public int getGold()

public class Location

public Location(int row, int col)
public int getRow()
public int getCol()

public class TreasureMap

private Treasure[][] map

public TreasureMap(int r, int c, ArrayList<Location> locs)
public int totalGold(Location start, Location end)
```

Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.

NO TEST MATERIAL ON THIS PAGE

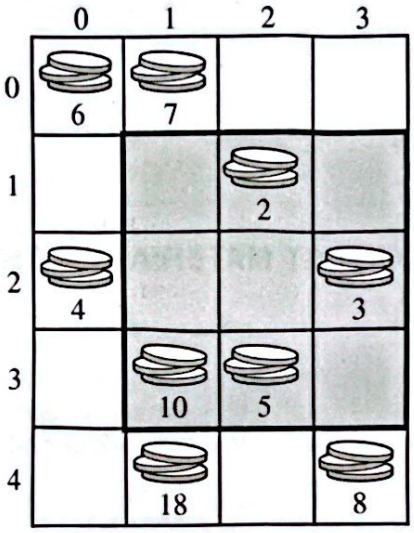
GO ON TO THE NEXT PAGE.

(b) Write the method `totalGold`. This method returns the total number of gold coins found in all `map` locations that are within the rectangular region defined by `start` and `end`, inclusive, where `start` is the upper left corner of the region and `end` is the lower right corner of the region. A null (empty) position in map contains no Treasure object. Assume that start and end are valid locations in map, the row index of start is less than or equal to the row index of end, and the column index of start is less than or equal to the column index of end.

Each of the following examples shows a representation of `map` containing `Treasure` objects. Numbers indicate the amount of gold in the object at a given location, and blanks denote `null` values.

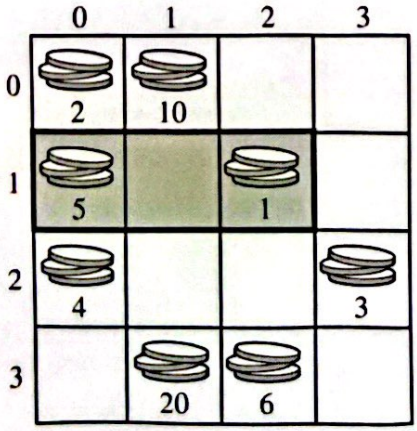
Example 1:

If `start` is `Location(1, 1)` and `end` is `Location(3, 3)`, then `totalGold(start, end)` should return `20` (2 + 3 + 10 + 5).



Example 2:

If `start` is `Location(1, 0)` and `end` is `Location(1, 2)`, then `totalGold(start, end)` should return `6` (5 + 1).



Unauthorized copying or reuse of any part of this page is illegal.

GO ON TO THE NEXT PAGE.