Section I

Multiple-Choice Questions

COMPUTER SCIENCE A

SECTION I

Time—1 hour and 15 minutes

Number of questions—40

Percent of total grade—50

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 box on the student answer sheet. No credit will be given for anything written in the examination booklet. Do not spend too much time on any one problem.

Notes:

- Assume that the classes listed in the Quick Reference found in the Appendix 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`.

GO ON TO THE NEXT PAGE.

1. Consider the following method.

```
public static int mystery(int[] arr)
{
  int x = 0;

  for (int k = 0; k < arr.length; k = k + 2)
    x = x + arr[k];

  return x;
}
```

Assume that the array `nums` has been declared and initialized as follows.

```
int[] nums = {3, 6, 1, 0, 1, 4, 2};
```

What value will be returned as a result of the call `mystery(nums)` ?

(A) 5

(B) 6

(C) 7

(D) 10

(E) 17

GO ON TO THE NEXT PAGE.

Questions 2-3 refer to the following information.

Consider the following partial class declaration.

```
public class SomeClass
{
  private int myA;
  private int myB;
  private int myC;

  // Constructor(s) not shown

  public int getA()
  {  return myA;  }

  public void setB(int value)
  {  myB = value;  }
}
```

2. The following declaration appears in another class.

```
SomeClass obj = new SomeClass();
```

Which of the following code segments will compile without error?

(A) `int x = obj.getA();`

(B)
```
int x;
obj.getA(x);
```

(C) `int x = obj.myA;`

(D) `int x = SomeClass.getA();`

(E) `int x = getA(obj);`

---

3. Which of the following changes to `SomeClass` will allow other classes to access but not modify the value of `myC` ?

(A) Make `myC` public.

(B) Include the method:
```
public int getC()
{  return myC;  }
```

(C) Include the method:
```
private int getC()
{  return myC;  }
```

(D) Include the method:
```
public void getC(int x)
{  x = myC;  }
```

(E) Include the method:
```
private void getC(int x)
{  x = myC;  }
```

GO ON TO THE NEXT PAGE.

4. Consider the following code segment.

```
int x = 7;
int y = 3;

if ((x < 10) && (y < 0))
  System.out.println("Value is: " + x * y);
else
  System.out.println("Value is: " + x / y);
```

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

(A) `Value is: 21`

(B) `Value is: 2.3333333`

(C) `Value is: 2`

(D) `Value is: 0`

(E) `Value is: 1`

GO ON TO THE NEXT PAGE.

5. Consider the following method.

```
public ArrayList<Integer> mystery(int n)
{
  ArrayList<Integer> seq = new ArrayList<Integer>();

  for (int k = 1; k <= n; k++)
    seq.add(new Integer(k * k + 3));

  return seq;
}
```

Which of the following is printed as a result of executing the following statement?

```
System.out.println(mystery(6));
```

(A) `[3, 4, 7, 12, 19, 28]`

(B) `[3, 4, 7, 12, 19, 28, 39]`

(C) `[4, 7, 12, 19, 28, 39]`

(D) `[39, 28, 19, 12, 7, 4]`

(E) `[39, 28, 19, 12, 7, 4, 3]`

6. Consider the following method that is intended to determine if the `double` values `d1` and `d2` are close enough to be considered equal. For example, given a `tolerance` of `0.001`, the values `54.32271` and `54.32294` would be considered equal.

```
/** @return true if d1 and d2 are within the specified tolerance,
 *          false otherwise
 */
public boolean almostEqual(double d1, double d2, double tolerance)
{
  /* missing code */
}
```

Which of the following should replace `/* missing code */` so that `almostEqual` will work as intended?

(A) `return (d1 - d2) <= tolerance;`

(B) `return ((d1 + d2) / 2) <= tolerance;`

(C) `return (d1 - d2) >= tolerance;`

(D) `return ((d1 + d2) / 2) >= tolerance;`

(E) `return Math.abs(d1 - d2) <= tolerance;`

GO ON TO THE NEXT PAGE.

7. Consider the following class declaration.

```
public class Person
{
  private String myName;
  private int myYearOfBirth;

  public Person(String name, int yearOfBirth)
  {
    myName = name;
    myYearOfBirth = yearOfBirth;
  }

  public String getName()
  {  return myName;  }

  public void setName(String name)
  {  myName = name;  }

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

Assume that the following declaration has been made.

```
Person student = new Person("Thomas", 1995);
```

Which of the following statements is the most appropriate for changing the name of `student` from `"Thomas"` to `"Tom"` ?

(A) `student = new Person("Tom", 1995);`

(B) `student.myName = "Tom";`

(C) `student.getName("Tom");`

(D) `student.setName("Tom");`

(E) `Person.setName("Tom");`

8. Consider the following class declaration.

```
public class Student
{
  private String myName;
  private int myAge;

  public Student()
  {  /* implementation not shown */  }

  public Student(String name, int age)
  {  /* implementation not shown */  }

  // No other constructors
}
```

Which of the following declarations will compile without error?

I. `Student a = new Student();`

II. `Student b = new Student("Juan", 15);`

III. `Student c = new Student("Juan", "15");`

(A) I only

(B) II only

(C) I and II only

(D) I and III only

(E) I, II, and III

9. Consider the following method that is intended to return the sum of the elements in the array `key`.

```
public static int sumArray(int[] key)
{
  int sum = 0;

  for (int i = 1; i <= key.length; i++)
  {
    /* missing code */
  }

  return sum;
}
```

Which of the following statements should be used to replace `/* missing code */` so that `sumArray` will work as intended?

(A) `sum = key[i];`

(B) `sum += key[i - 1];`

(C) `sum += key[i];`

(D) `sum += sum + key[i - 1];`

(E) `sum += sum + key[i];`

Questions 10-11 refer to the following information.

Consider the following instance variable and methods. You may assume that `data` has been initialized with length > 0. The methods are intended to return the index of an array element equal to `target`, or -1 if no such element exists.

```
private int[] data;

public int seqSearchRec(int target)
{
  return seqSearchRecHelper(target, data.length - 1);
}


private int seqSearchRecHelper(int target, int last)
{
  // Line 1

  if (data[last] == target)
    return last;
  else
    return seqSearchRecHelper(target, last - 1);
}
```

10. For which of the following test cases will the call `seqSearchRec(5)` always result in an error?

I. `data` contains only one element.

II. `data` does not contain the value 5.

III. `data` contains the value 5 multiple times.

(A) I only

(B) II only

(C) III only

(D) I and II only

(E) I, II, and III

---

11. Which of the following should be used to replace `// Line 1` in `seqSearchRecHelper` so that `seqSearchRec` will work as intended?

(A)
```
if (last <= 0)
  return -1;
```

(B)
```
if (last < 0)
  return -1;
```

(C)
```
if (last < data.length)
  return -1;
```

(D) `while (last < data.length)`

(E) `while (last >= 0)`

GO ON TO THE NEXT PAGE.

12. Consider the following method.

```
public String mystery(String input)
{
  String output = "";

  for (int k = 1; k < input.length(); k = k + 2)
  {
    output += input.substring(k, k + 1);
  }

  return output;
}
```

What is returned as a result of the call `mystery("computer")` ?

(A) `"computer"`

(B) `"cmue"`

(C) `"optr"`

(D) `"ompute"`

(E) Nothing is returned because an `IndexOutOfBoundsException` is thrown.

GO ON TO THE NEXT PAGE.

13. Consider the following code segment.

```
int[] arr = {7, 2, 5, 3, 0, 10};
for (int k = 0; k < arr.length - 1; k++)
{
  if (arr[k] > arr[k + 1])
    System.out.print(k + "  " + arr[k] + "  ");
}
```

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

(A) `0  2  2  3  3  0`

(B) `0  7  2  5  3  3`

(C) `0  7  2  5  5  10`

(D) `1  7  3  5  4  3`

(E) `7  2  5  3  3  0`

GO ON TO THE NEXT PAGE.

14. Consider the following interface and class declarations.

```
public interface Vehicle
{
  /** @return the mileage traveled by this Vehicle
   */
  double getMileage();
}


public class Fleet
{
  private ArrayList<Vehicle> myVehicles;

  /** @return the mileage traveled by all vehicles in this Fleet
   */
  public double getTotalMileage()
  {
    double sum = 0.0;

    for (Vehicle v : myVehicles)
    {
      sum += /* expression */ ;
    }

    return sum;
  }

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

Which of the following can be used to replace `/* expression */` so that `getTotalMileage` returns the total of the miles traveled for all vehicles in the fleet?

(A) `getMileage(v)`

(B) `myVehicles[v].getMileage()`

(C) `Vehicle.get(v).getMileage()`

(D) `myVehicles.get(v).getMileage()`

(E) `v.getMileage()`

15. Consider the following method, `isSorted`, which is intended to return `true` if an array of integers is sorted in nondecreasing order and to return `false` otherwise.

```
/** @param data  an array of integers
 *  @return true  if the values in the array appear in sorted (nondecreasing) order
 */
public static boolean isSorted(int[] data)
{
  /* missing code */
}
```

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

I.
```
for (int k = 1; k < data.length; k++)
{
  if (data[k - 1] > data[k])
    return false;
}
return true;
```

II.
```
for (int k = 0; k < data.length; k++)
{
  if (data[k] > data[k + 1])
    return false;
}
return true;
```

III.
```
for (int k = 0; k < data.length - 1; k++)
{
  if (data[k] > data[k + 1])
    return false;
  else
    return true;
}
return true;
```

(A) I only

(B) II only

(C) III only

(D) I and II only

(E) I and III only

16. Consider the following incomplete method that is intended to return an array that contains the contents of its first array parameter followed by the contents of its second array parameter.

```
public static int[] append(int[] a1, int[] a2)
{
  int[] result = new int[a1.length + a2.length];

  for (int j = 0; j < a1.length; j++)
    result[j] = a1[j];

  for (int k = 0; k < a2.length; k++)
    result[ /* index */ ] = a2[k];

  return result;
}
```

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

(A) `j`

(B) `k`

(C) `k + a1.length - 1`

(D) `k + a1.length`

(E) `k + a1.length + 1`

17. Consider the following code segment.

```
int[] arr = {1, 2, 3, 4, 5, 6, 7};

for (int k = 3; k < arr.length - 1; k++)
  arr[k] = arr[k + 1];
```

Which of the following represents the contents of `arr` as a result of executing the code segment?

(A) `{1, 2, 3, 4, 5, 6, 7}`

(B) `{1, 2, 3, 5, 6, 7}`

(C) `{1, 2, 3, 5, 6, 7, 7}`

(D) `{1, 2, 3, 5, 6, 7, 8}`

(E) `{2, 3, 4, 5, 6, 7, 7}`

GO ON TO THE NEXT PAGE.

18. Assume that `myList` is an `ArrayList` that has been correctly constructed and populated with objects. Which of the following expressions produces a valid random index for `myList` ?

(A) `(int)( Math.random() * myList.size() ) - 1`

(B) `(int)( Math.random() * myList.size() )`

(C) `(int)( Math.random() * myList.size() ) + 1`

(D) `(int)( Math.random() * (myList.size() + 1) )`

(E) `Math.random(myList.size())`

GO ON TO THE NEXT PAGE.

19. Assume that `a` and `b` have been defined and initialized as `int` values. The expression

```
!(!(a != b ) && (b > 7))
```

is equivalent to which of the following?

(A) `(a != b) || (b < 7)`

(B) `(a != b) || (b <= 7)`

(C) `(a == b) || (b <= 7)`

(D) `(a != b) && (b <= 7)`

(E) `(a == b) && (b > 7)`

GO ON TO THE NEXT PAGE.

20. Consider the following method.

```
public static void arrayMethod(int nums[])
{
  int j = 0;
  int k = nums.length - 1;

  while (j < k)
  {
    int x = nums[j];
    nums[j] = nums[k];
    nums[k] = x;
    j++;
    k--;
  }
}
```

Which of the following describes what the method `arrayMethod()` does to the array `nums`?

(A) The array `nums` is unchanged.

(B) The first value in `nums` is copied to every location in the array.

(C) The last value in `nums` is copied to every location in the array.

(D) The method generates an `ArrayIndexOutOfBoundsException`.

(E) The contents of the array `nums` are reversed.

Questions 21-25 refer to the code from the GridWorld case study. A copy of the code is provided in the Appendix.

21. Consider the design of a `Grasshopper` class that extends `Bug`. When asked to move, a `Grasshopper` moves to a randomly chosen empty adjacent location that is within the grid. If there is no empty adjacent location that is within the grid, the `Grasshopper` does not move, but turns 45 degrees to the right without changing its location.

    Which method(s) of the `Bug` class should the `Grasshopper` class override so that a `Grasshopper` can behave as described above?

    I. `act()`

    II. `move()`

    III. `canMove()`

    (A) I only

    (B) II only

    (C) I and II only

    (D) II and III only

    (E) I, II, and III

GO ON TO THE NEXT PAGE.

22. Assume that gus has been defined and initialized as a Bug object in a class that contains the following code segment.

```
int numTurnsMade = 0;
for (int k = 1; k <= 100; k++)
{
  int dir = gus.getDirection();
  int dirTurn = dir + Location.HALF_RIGHT;
  gus.act();
  if ( /* expression */ )
    numTurnsMade++;
}
```

Which of the following could be used to replace /* *expression* */ so that the variable numTurnsMade accurately stores the number of times that gus turns 45 degrees to the right?

(A) dir == dirTurn

(B) dir == gus.getDirection()

(C) dirTurn == Location.HALF_RIGHT

(D) dirTurn == gus.getDirection()

(E) Location.HALF_RIGHT == gus.getDirection()

GO ON TO THE NEXT PAGE.

23. Consider the following method that is intended to return an `ArrayList` of all the locations in `grd` that contain actors facing in direction `dir`.

```
public ArrayList<Location> findLocsFacingDir(int dir, Grid<Actor> grd)
{
  ArrayList<Location> desiredLocs = new ArrayList<Location>();

  for (Location loc : grd.getOccupiedLocations())
  {
    if ( /* expression */ == dir )
      desiredLocs.add(loc);
  }
  return desiredLocs;
}
```

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

(A) `loc.getDirection()`

(B) `getDirection(loc)`

(C) `((Actor) loc).getDirection()`

(D) `grd(loc).getDirection()`

(E) `grd.get(loc).getDirection()`

GO ON TO THE NEXT PAGE.

24. A `ColorChangingCritter` behaves like a `ChameleonCritter` but does not turn when it moves. A partial declaration for the `ColorChangingCritter` class is as follows.

```
public class ColorChangingCritter extends ChameleonCritter
{
  public void makeMove(Location loc)
  {  /* missing code */  }
}
```

Which of the following replacements for `/* missing code */` will correctly implement the desired behavior?

I. `moveTo(loc);`

II. `super.super.makeMove(loc);`

III.
```
int dir = getDirection();
super.makeMove(loc);
setDirection(dir);
```

(A) I only

(B) III only

(C) I and II only

(D) I and III only

(E) I, II, and III

25. A `MunchingCritter` acts by selecting one adjacent actor of any type, eating it (removing it from the grid), and moving to occupy its location. If there is no adjacent actor, the `MunchingCritter` moves like a normal critter. Consider the following three implementations of `MunchingCritter`.

Implementation I

```
public class MunchingCritter extends Critter
{
  private Location eatLoc;   // Remember location of critter that was eaten

  public void processActors(ArrayList<Actor> actors)
  {
    if (actors.size() == 0)
      eatLoc = null;
    else
    {
      Actor selected = actors.get(0);
      eatLoc = selected.getLocation();
      selected.removeSelfFromGrid();
    }
  }

  public Location selectMoveLocation(ArrayList<Location> locs)
  {
    if (eatLoc == null)
      return super.selectMoveLocation(locs);
    else
      return eatLoc;
  }
}
```

Implementation II

```
public class MunchingCritter extends Critter
{
  private Location eatLoc;   // Remember location of critter that was eaten

  public void processActors(ArrayList<Actor> actors)
  {
    if (actors.size() == 0)
      eatLoc = null;
    else
    {
      Actor selected = actors.get(0);
      eatLoc = selected.getLocation();
      selected.removeSelfFromGrid();
    }
  }

  public void makeMove(Location loc)
  {
    if (eatLoc == null)
      moveTo(loc);
    else
      moveTo(eatLoc);
  }
}
```

Implementation III

```
public class MunchingCritter extends Critter
{
  private boolean hasEaten;   // Remember if this critter ate something during this step

  public void processActors(ArrayList<Actor> actors)
  {
    if (actors.size() == 0)
      hasEaten = false;
    else
    {
      Actor selected = actors.get(0);
      Location moveLoc = selected.getLocation();
      selected.removeSelfFromGrid();
      moveTo(moveLoc);
      hasEaten = true;
    }
  }

  public void makeMove(Location loc)
  {
    if (!hasEaten)
      moveTo(loc);
  }
}
```

Which of the implementations would be considered to be well designed, in that they satisfy the postconditions in `Critter.java` ?

(A) I only

(B) II only

(C) III only

(D) I and II only

(E) I, II, and III

GO ON TO THE NEXT PAGE.

26. Assume that the array `arr` has been defined and initialized as follows.

```
int[] arr = /* initial values for the array */ ;
```

Which of the following will correctly print all of the odd integers contained in `arr` but none of the even integers contained in `arr` ?

(A)
```
for (int x : arr)
  if (x % 2 == 1)
    System.out.println(x);
```

(B)
```
for (int k = 1; k < arr.length; k++)
  if (arr[k] % 2 == 1)
    System.out.println(arr[k]);
```

(C)
```
for (int x : arr)
  if (x % 2 == 1)
    System.out.println(arr[x]);
```

(D)
```
for (int k = 0; k < arr.length; k++)
  if (arr[k] % 2 == 1)
    System.out.println(k);
```

(E)
```
for (int x : arr)
  if (arr[x] % 2 == 1)
    System.out.println(arr[x]);
```

GO ON TO THE NEXT PAGE.

Questions 27-28 refer to the following method.

```
public static int mystery(int n)
{
  int x = 1;
  int y = 1;

  // Point A

  while (n > 2)
  {
    x = x + y;

    // Point B

    y = x - y;
    n--;
  }

  // Point C

  return x;
}
```

27. What value is returned as a result of the call `mystery(6)` ?

(A) 1

(B) 5

(C) 6

(D) 8

(E) 13

---

28. Which of the following is true of method `mystery` ?

(A) `x` will sometimes be `1` at `// Point B`.

(B) `x` will never be `1` at `// Point C`.

(C) `n` will never be greater than `2` at `// Point A`.

(D) `n` will sometimes be greater than `2` at `// Point C`.

(E) `n` will always be greater than `2` at `// Point B`.

GO ON TO THE NEXT PAGE.

29. Consider the following code segment.

```
for (int k = 1; k <= 100; k++)
  if ((k % 4) == 0)
    System.out.println(k);
```

Which of the following code segments will produce the same output as the code segment above?

(A)
```
for (int k = 1; k <= 25; k++)
  System.out.println(k);
```

(B)
```
for (int k = 1; k <= 100; k = k + 4)
  System.out.println(k);
```

(C)
```
for (int k = 1; k <= 100; k++)
  System.out.println(k % 4);
```

(D)
```
for (int k = 4; k <= 25; k = 4 * k)
  System.out.println(k);
```

(E)
```
for (int k = 4; k <= 100; k = k + 4)
  System.out.println(k);
```

30. Consider the following method.

```
public static String scramble(String word, int howFar)
{
  return word.substring(howFar + 1, word.length()) +
         word.substring(0, howFar);
}
```

What value is returned as a result of the call `scramble("compiler", 3)`?

(A) "compiler"

(B) "pilercom"

(C) "ilercom"

(D) "ilercomp"

(E) No value is returned because an IndexOutOfBoundsException will be thrown.

31. Consider the following method.

```
public void mystery(int[] data)
{
  for (int k = 0; k < data.length - 1; k++)
    data[k + 1] = data[k] + data[k + 1];
}
```

The following code segment appears in another method in the same class.

```
int[] values = {5, 2, 1, 3, 8};
mystery(values);
for (int v : values)
  System.out.print(v + " ");
System.out.println();
```

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

(A) `5 2 1 3 8`

(B) `5 7 3 4 11`

(C) `5 7 8 11 19`

(D) `7 3 4 11 8`

(E) Nothing is printed because an `ArrayIndexOutOfBoundsException` is thrown during the execution of method `mystery`.

GO ON TO THE NEXT PAGE.

32. Consider the following method.

```
public int compute(int n, int k)
{
  int answer = 1;

  for (int i = 1; i <= k; i++)
    answer *= n;

  return answer;
}
```

Which of the following represents the value returned as a result of the call `compute(n, k)` ?

(A) `n*k`

(B) `n!`

(C) n^k

(D) 2^k

(E) k^n

GO ON TO THE NEXT PAGE.

33. Consider the following code segment.

```
int sum = 0;
int k = 1;
while (sum < 12 || k < 4)
  sum += k;

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

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

(A) 6

(B) 10

(C) 12

(D) 15

(E) Nothing is printed due to an infinite loop.

34. Consider the following class declarations.

```
public class Point
{
  private double x;   // x-coordinate
  private double y;   // y-coordinate

  public Point()
  {
    x = 0;
    y = 0;
  }

  public Point(double a, double b)
  {
    x = a;
    y = b;
  }

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


public class Circle
{
  private Point center;
  private double radius;

  /** Constructs a circle where (a, b) is the center and r is the radius.
   */
  public Circle(double a, double b, double r)
  {
    /* missing code */
  }
}
```

Which of the following replacements for `/* missing code */` will correctly implement the `Circle` constructor?

I.
```
center = new Point();
radius = r;
```

II.
```
center = new Point(a, b);
radius = r;
```

III.
```
center = new Point();
center.x = a;
center.y = b;
radius = r;
```

(A) I only

(B) II only

(C) III only

(D) II and III only

(E) I, II, and III

35. Consider the following code segment.

```
int num = 2574;
int result = 0;

while (num > 0)
{
  result = result * 10 + num % 10;
  num /= 10;
}
System.out.println(result);
```

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

(A) 2

(B) 4

(C) 18

(D) 2574

(E) 4752

GO ON TO THE NEXT PAGE.

36. Consider the following method.

```
public void test(int x)
{
  int y;

  if (x % 2 == 0)
    y = 3;
  else if (x > 9)
    y = 5;
  else
    y = 1;

  System.out.println("y = " + y);
}
```

Which of the following test data sets would test each possible output for the method?

(A) 8, 9, 12

(B) 7, 9, 11

(C) 8, 9, 11

(D) 8, 11, 13

(E) 7, 9, 10

37. Consider the following code segment.

```
int x = 1;
while ( /* missing code */ )
{
  System.out.print(x + " ");
  x = x + 2;
}
```

Consider the following possible replacements for `/* missing code */`.

I. `x < 6`

II. `x != 6`

III. `x < 7`

Which of the proposed replacements for `/* missing code */` will cause the code segment to print only the values `1 3 5` ?

(A) I only

(B) II only

(C) I and II only

(D) I and III only

(E) I, II, and III

GO ON TO THE NEXT PAGE.

38. Assume that `x` and `y` have been declared and initialized with `int` values. Consider the following Java expression.

```
(y > 10000) || (x > 1000 && x < 1500)
```

Which of the following is equivalent to the expression given above?

(A) `(y > 10000 || x > 1000) && (y > 10000 || x < 1500)`

(B) `(y > 10000 || x > 1000) || (y > 10000 || x < 1500)`

(C) `(y > 10000) && (x > 1000 || x < 1500)`

(D) `(y > 10000 && x > 1000) || (y > 10000 && x < 1500)`

(E) `(y > 10000 && x > 1000) && (y > 10000 && x < 1500)`

GO ON TO THE NEXT PAGE.

39. Consider the following recursive method.

```
public int recur(int n)
{
  if (n <= 10)
    return n * 2;
  else
    return recur(recur(n / 3));
}
```

What value is returned as a result of the call `recur(27)` ?

(A) 8

(B) 9

(C) 12

(D) 16

(E) 18

40. Consider the following recursive method.

```
public static void whatsItDo(String str)
{
  int len = str.length();
  if (len > 1)
  {
    String temp = str.substring(0, len - 1);
    whatsItDo(temp);
    System.out.println(temp);
  }
}
```

What is printed as a result of the call `whatsItDo("WATCH")` ?

(A)
```
WATC
WAT
WA
W
```

(B)
```
WATCH
WATC
WAT
WA
```

(C)
```
W
WA
WAT
WATC
```

(D)
```
W
WA
WAT
WATC
WATCH
```

(E)
```
WATCH
WATC
WAT
WA
W
WA
WAT
WATC
WATCH
```

END OF SECTION I

IF YOU FINISH BEFORE TIME IS CALLED,
YOU MAY CHECK YOUR WORK ON THIS SECTION.

DO NOT GO ON TO SECTION II UNTIL YOU ARE TOLD TO DO SO.