AP Computer Science A: Practice Exam 1

Part I (Multiple Choice)

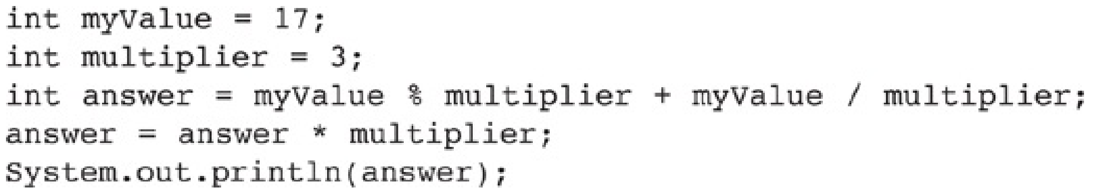
Time: 90 minutes
Number of questions: 40
Percent of total score: 50

Directions: Choose the best answer for each problem. Some problems take longer than others. Consider how much time you have left before spending too much time on any one problem.

Notes:

- You may assume all import statements have been included where they are needed.
- You may assume that the parameters in method calls are not null.
- You may assume that declarations of variables and methods appear within the context of an enclosing class.

1. Consider the following code segment.

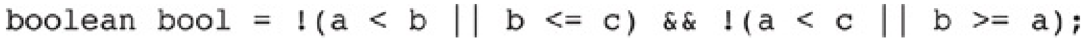


```
int myValue = 17;
int multiplier = 3;
int answer = myValue % multiplier + myValue / multiplier;
answer = answer * multiplier;
System.out.println(answer);
```

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

(A) `9`
(B) `10`
(C) `7`
(D) `30`
(E) `21`

2. Assume that `a`, `b`, and `c` have been declared and correctly initialized with `int` values. Consider the following expression.

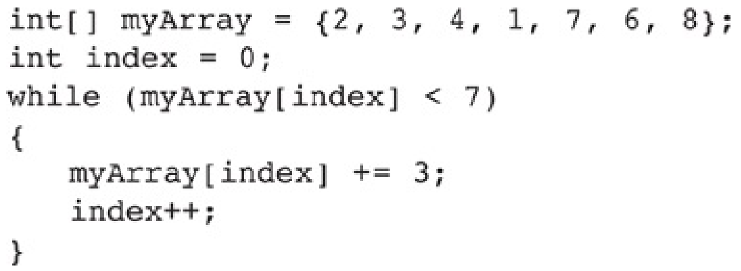


```
boolean bool = !(a < b || b <= c) && !(a < c || b >= a);
```

Under what conditions does `bool` evaluate to true?

(A) `a = 1`, `b = 2`, `c = 3`
(B) `a = 3`, `b = 2`, `c = 1`
(C) `a = 3`, `b = 1`, `c = 2`
(D) All conditions; `bool` is always `true`.
(E) No conditions; `bool` is always `false`.

3. Consider the following code segment.

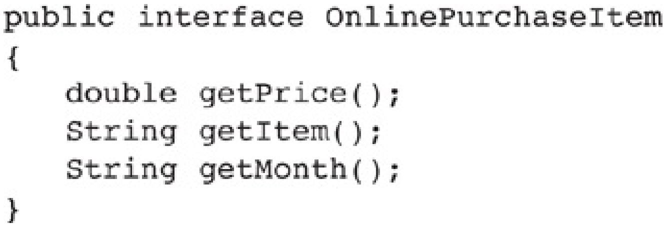


```
int[] myArray = {2, 3, 4, 1, 7, 6, 8};
int index = 0;
while (myArray[index] < 7)
{
   myArray[index] += 3;
   index++;
}
```

What values are stored in `myArray` after executing the code segment?

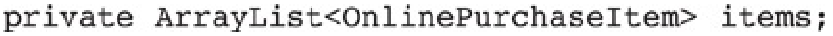
(A) `{2, 3, 4, 1, 7, 6, 8}`
(B) `{2, 3, 4, 1, 7, 9, 11}`
(C) `{5, 6, 7, 4, 7, 9, 8}`
(D) `{5, 6, 7, 4, 7, 6, 8}`
(E) `{5, 6, 7, 4, 0, 9, 11}`

4. Consider the following interface used by a company to represent the items it has available for online purchase.



```
public interface OnlinePurchaseItem
{
   double getPrice();
   String getItem();
   String getMonth();
}
```

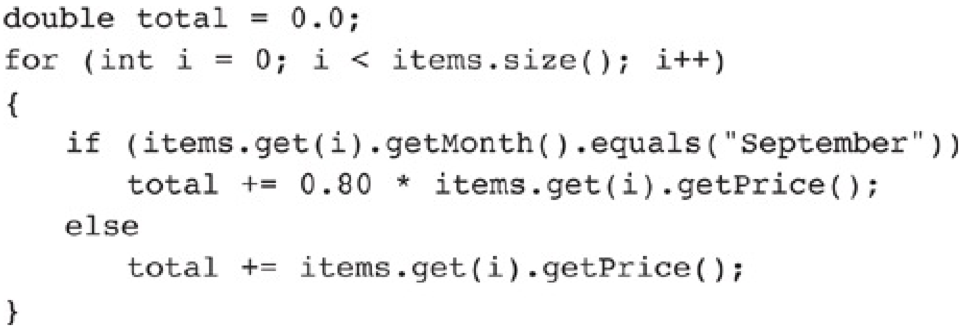
The company bills at the end of the quarter. Until then, it uses an `ArrayList` of `OnlinePurchaseItem` objects to track a customer’s purchases.



```
private ArrayList<OnlinePurchaseItem> items;
```

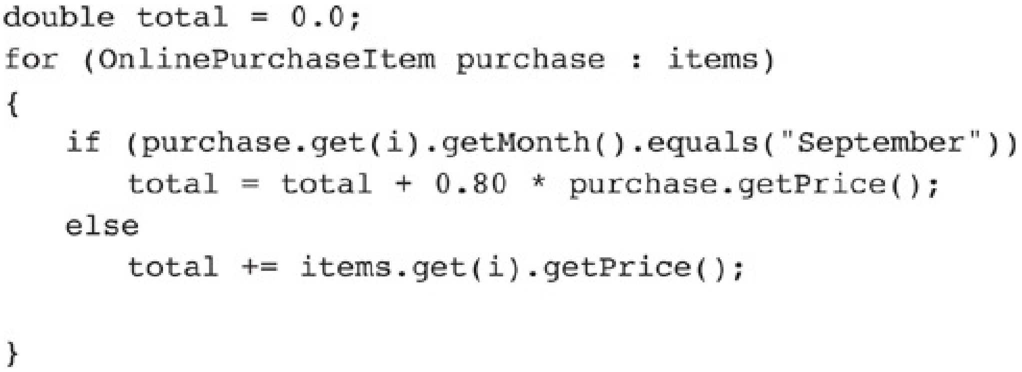
The company decides to offer a 20 percent off promotion on all items purchased in September. Which of the following code segments properly calculates the correct total price at the end of the quarter?

I.



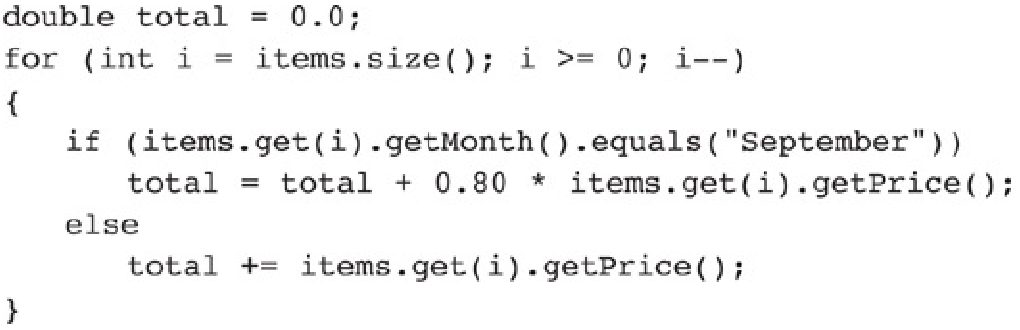
```
double total = 0.0;
for (int i = 0; i < items.size(); i++)
{
   if (items.get(i).getMonth().equals("September"))
      total += 0.80 * items.get(i).getPrice();
   else
      total += items.get(i).getPrice();
}
```

II.



```
double total = 0.0;
for (OnlinePurchaseItem purchase : items)
{
   if (purchase.get(i).getMonth().equals("September"))
      total = total + 0.80 * purchase.getPrice();
   else
      total += items.get(i).getPrice();


}
```

III.



```
double total = 0.0;
for (int i = items.size(); i >= 0; i--)
{
   if (items.get(i).getMonth().equals("September"))
      total = total + 0.80 * items.get(i).getPrice();
   else
      total += items.get(i).getPrice();
}
```

(A) I only

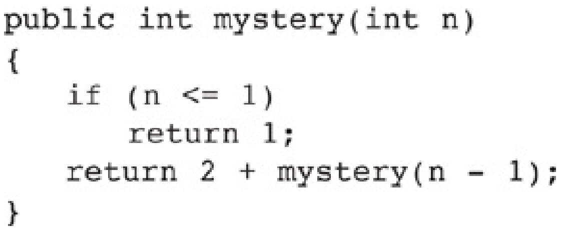
(B) II only

(C) I and II only

(D) II and III only

(E) I, II, and III

5. Consider the following method.



```
public int mystery(int n)
{
   if (n <= 1)
      return 1;
   return 2 + mystery(n - 1);
}
```

What value is returned by the call `mystery(5)`?

(A) `1`

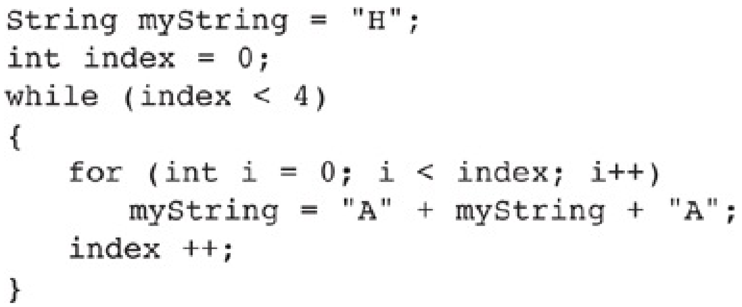
(B) `7`

(C) `8`

(D) `9`

(E) `10`

6. Consider the following code segment.



```
String myString = "H";
int index = 0;
while (index < 4)
{
   for (int i = 0; i < index; i++)
      myString = "A" + myString + "A";
   index ++;
}
```

What is the value of `myString` after executing the code segment?

(A) `"H"`

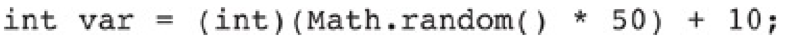
(B) `"AHA"`

(C) `"AAAHAAA"`

(D) `"AAAAHAAAA"`

(E) `"AAAAAAHAAAAAA"`

7. Consider the following statement.



```
int var = (int)(Math.random() * 50) + 10;
```

What are the possible values of `var` after executing the statement?

(A) All integers from 1 to 59 (inclusive)

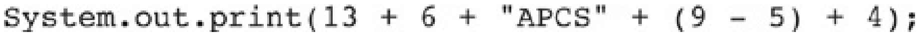
(B) All integers from 10 to 59 (inclusive)

(C) All integers from 10 to 60 (inclusive)

(D) All real numbers from 50 to 60 (not including 60)

(E) All real numbers from 10 to 60 (not including 60)

8. Consider the following statement.

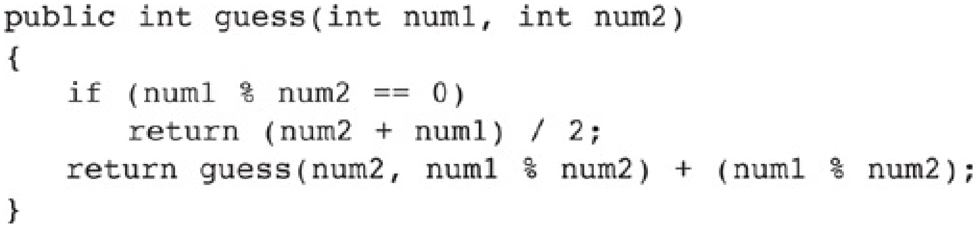


```
System.out.print(13 + 6 + "APCS" + (9 - 5) + 4);
```

What is printed as a result of executing the statement?

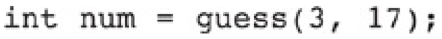
(A) `136APCS(9 - 5)4`
(B) `19APCS8`
(C) `19APCS44`
(D) `136APCS8`
(E) `136APCS44`

9. Consider the following method.



```
public int guess(int num1, int num2)
{
   if (num1 % num2 == 0)
      return (num2 + num1) / 2;
   return guess(num2, num1 % num2) + (num1 % num2);
}
```

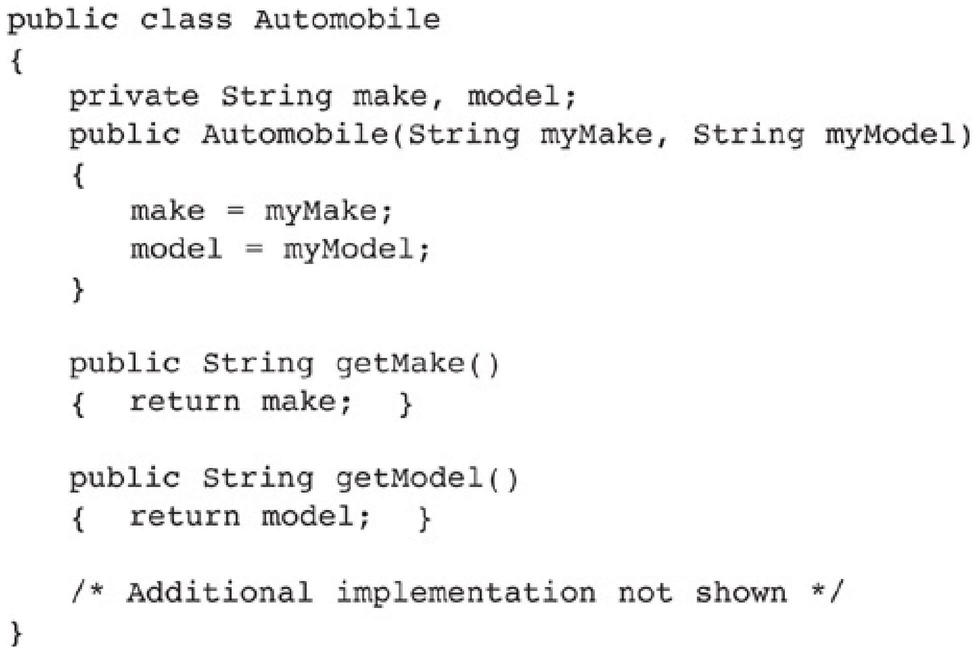
What is the value of `num` after executing the following code statement?



```
int num = guess(3, 17);
```

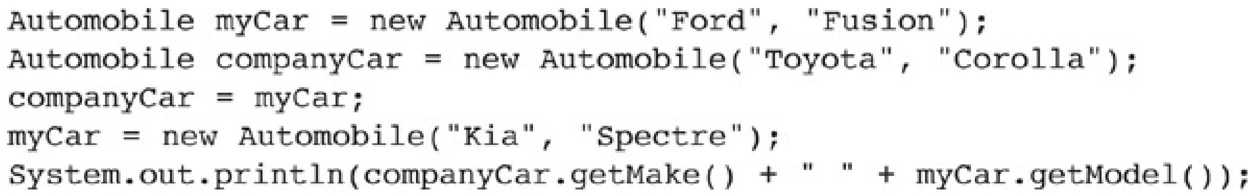
(A) `6`
(B) `7`
(C) `10`
(D) Nothing is returned. Modulus by 0 causes an `ArithmeticException`.
(E) Nothing is returned. Infinite recursion causes a stack overflow error.

10. Consider the following class.



```
public class Automobile
{
   private String make, model;
   public Automobile(String myMake, String myModel)
   {
      make = myMake;
      model = myModel;
   }

   public String getMake()
   {  return make;  }

   public String getModel()
   {  return model;  }

   /* Additional implementation not shown */
}
```

Consider the following code segment that appears in another class.



```
Automobile myCar = new Automobile("Ford", "Fusion");
Automobile companyCar = new Automobile("Toyota", "Corolla");
companyCar = myCar;
myCar = new Automobile("Kia", "Spectre");
System.out.println(companyCar.getMake() + " " + myCar.getModel());
```

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

(A) `Ford Spectre`

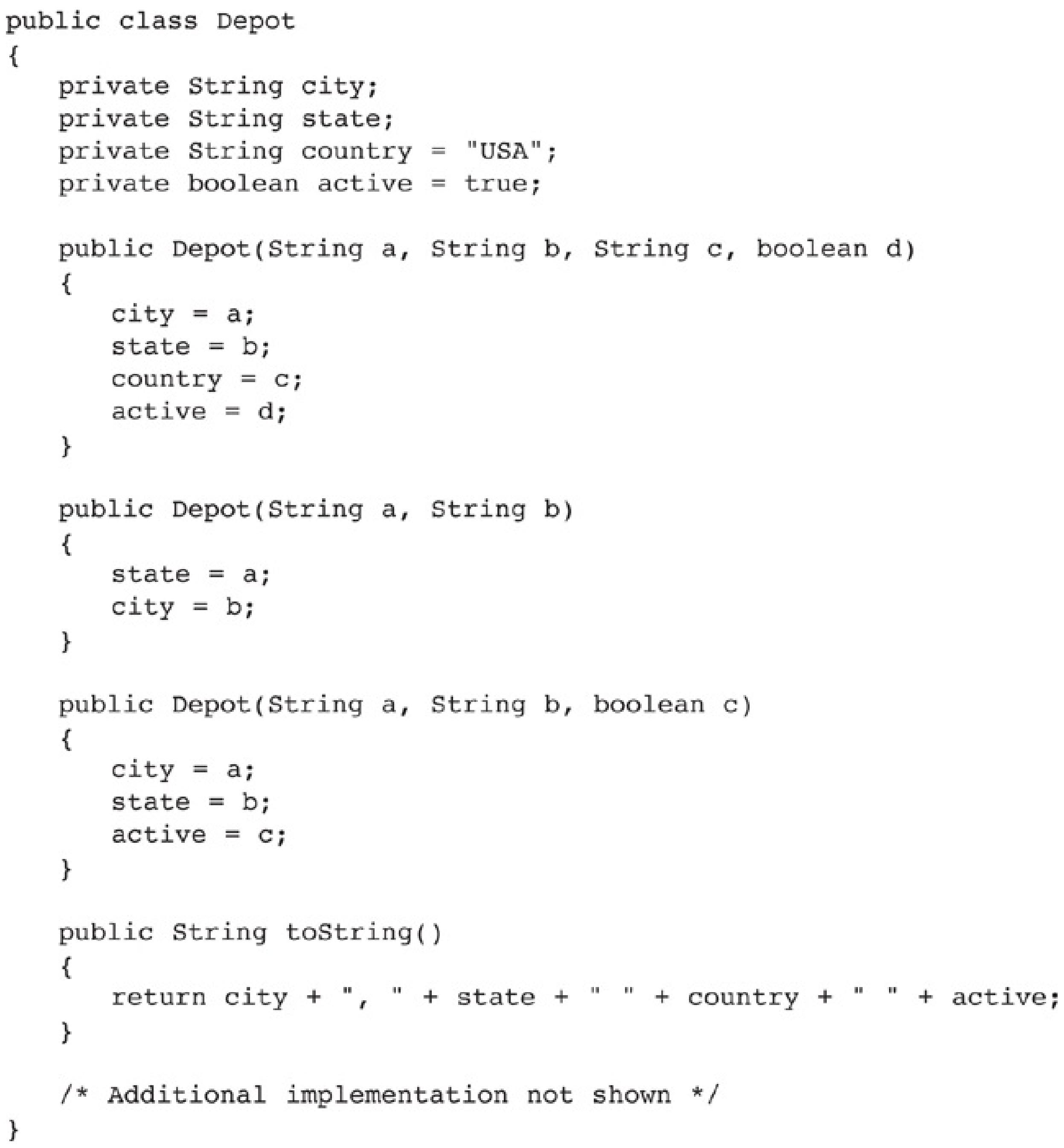
(B) `Ford Fusion`

(C) `Kia Spectre`

(D) `Toyota Corolla`

(E) `Toyota Spectre`

<u>Questions 11–14 refer to the following class definition.</u>



```
public class Depot
{
   private String city;
   private String state;
   private String country = "USA";
   private boolean active = true;

   public Depot(String a, String b, String c, boolean d)
   {
      city = a;
      state = b;
      country = c;
      active = d;
   }

   public Depot(String a, String b)
   {
      state = a;
      city = b;
   }

   public Depot(String a, String b, boolean c)
   {
      city = a;
      state = b;
      active = c;
   }

   public String toString()
   {
      return city + ", " + state + " " + country + " " + active;
   }

   /* Additional implementation not shown */
}
```

11. The `Depot` class has three constructors. Which of the following is the correct term for this practice?

(A) Overriding
(B) Procedural abstraction
(C) Encapsulation
(D) Polymorphism
(E) Overloading

12. Consider the following code segment in another class.

```
Depot station = new Depot("Oakland", "California");
System.out.println(station);
```

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

(A) `California, Oakland USA true`

(B) `California, Oakland USA active`

(C) `USA, California USA true`

(D) `Oakland, California USA active`

(E) `Oakland, California USA true`

13. Consider the following class definition.

```
public class WhistleStop extends Depot
```

Which of the following constructors compiles without error?

I.
```
public WhistleStop()
{
   super();
}
```

II.
```
public WhistleStop()
{
   super("Waubaushene", "Ontario", "Canada", true);
}
```

III.
```
public WhistleStop(String city, String province)
{
   super(city, province);
}
```

(A) I only

(B) II only

(C) III only

(D) II and III only

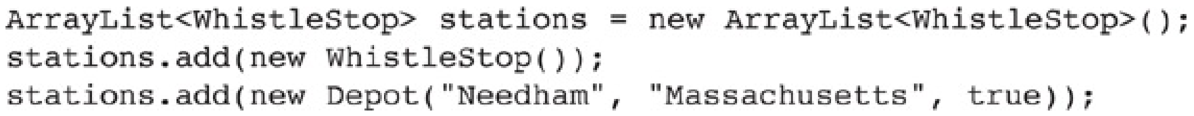
(E) I, II, and III

14. Assume a correct no-argument constructor has been added to the `WhistleStop` class.

Which of the following code segments compiles without error?

I.
```
ArrayList<Depot> stations = new ArrayList<Depot>();
stations.add(new WhistleStop());
stations.add(new Depot("Mansonville", "Quebec", "Canada", false));
```

II.
```
ArrayList<WhistleStop> stations = new ArrayList<Depot>();
stations.add(new WhistleStop());
stations.add(new Depot("Orinda", "California", true));
```

III. 

```
ArrayList<WhistleStop> stations = new ArrayList<WhistleStop>();
stations.add(new WhistleStop());
stations.add(new Depot("Needham", "Massachusetts", true));
```

(A) I only

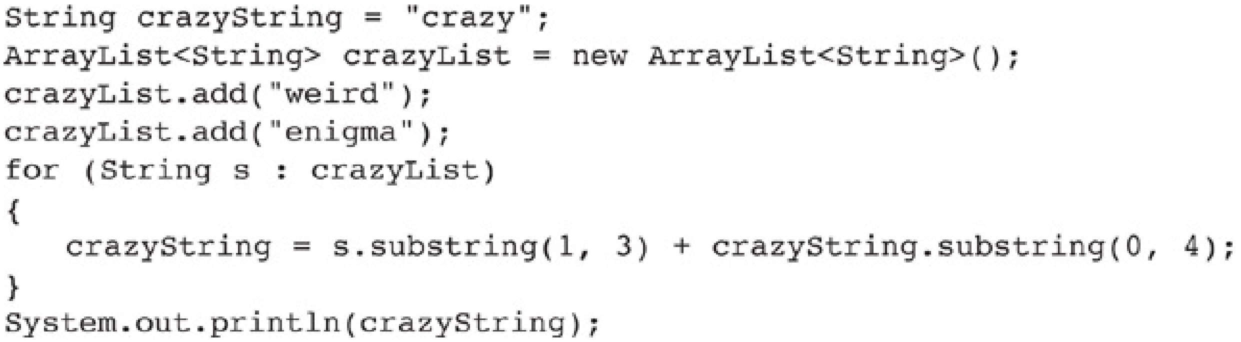
(B) I and II only

(C) I and III only

(D) II and III only

(E) I, II, and III

15. Consider the following code segment.



```
String crazyString = "crazy";
ArrayList<String> crazyList = new ArrayList<String>();
crazyList.add("weird");
crazyList.add("enigma");
for (String s : crazyList)
{
   crazyString = s.substring(1, 3) + crazyString.substring(0, 4);
}
System.out.println(crazyString);
```

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

(A) `nicraz`

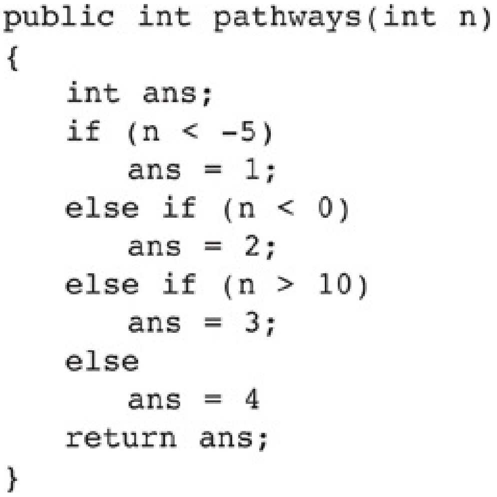
(B) `nieicr`

(C) `nieicraz`

(D) `einicraz`

(E) `weienicraz`

16. Consider the following method.



```
public int pathways(int n)
{
   int ans;
   if (n < -5)
      ans = 1;
   else if (n < 0)
      ans = 2;
   else if (n > 10)
      ans = 3;
   else
      ans = 4
   return ans;
}
```

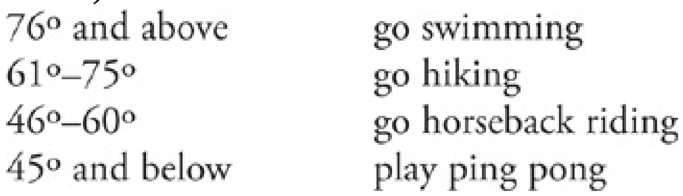
Which of the following sets of data tests every possible path through the code?

(A) `-6, -1, 15, 12`

(B) `-5, -3, 12, 15`
(C) `-8, -5, 8, 10`
(D) `-6, 0, 20, 7`
(E) `-10, -5, 10, 12`

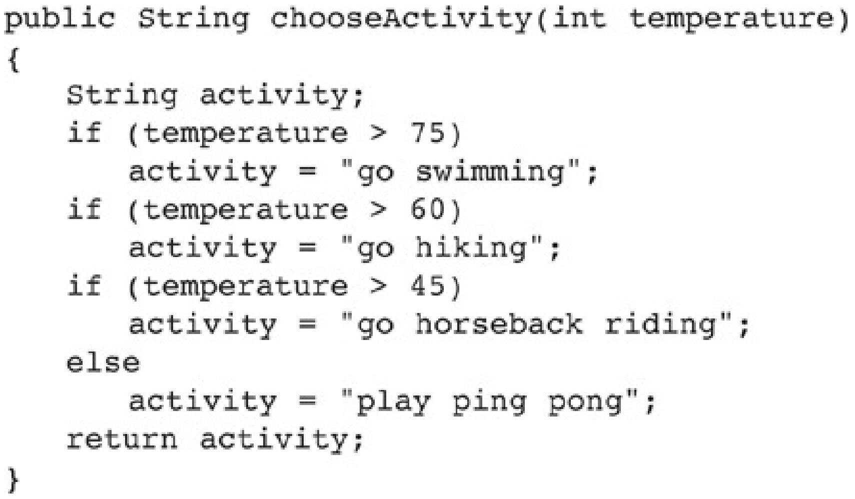
Questions 17–18 refer to the following scenario.

A resort wants to recommend activities for its guests, based on the temperature (degrees F) as follows:



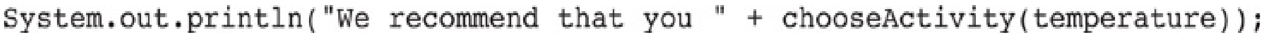
76° and above	go swimming
61°–75°	go hiking
46°–60°	go horseback riding
45° and below	play ping pong

17. Consider the following method.



```
public String chooseActivity(int temperature)
{
   String activity;
   if (temperature > 75)
      activity = "go swimming";
   if (temperature > 60)
      activity = "go hiking";
   if (temperature > 45)
      activity = "go horseback riding";
   else
      activity = "play ping pong";
   return activity;
}
```

Consider the following statement.

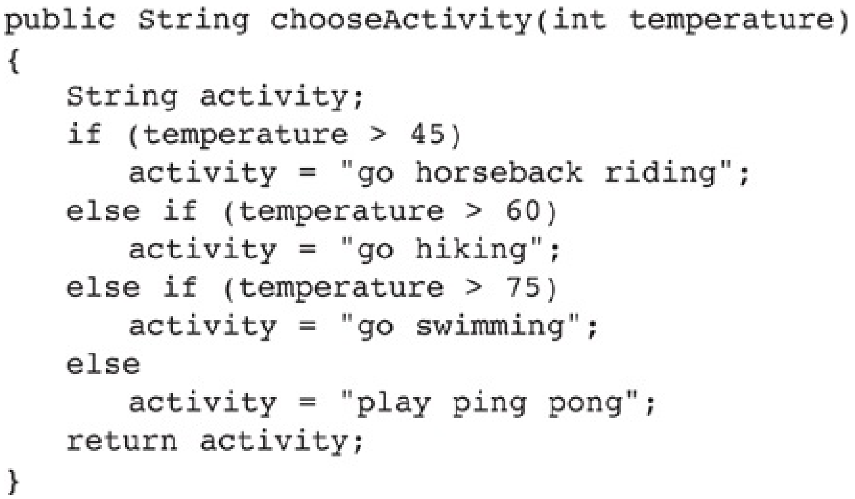


```
System.out.println("We recommend that you " + chooseActivity(temperature));
```

For which temperature range is the correct suggestion printed (as defined above)?

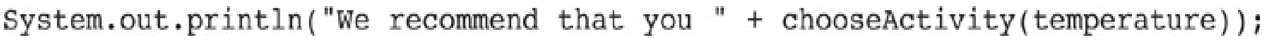
(A) `temperature > 75`
(B) `temperature > 60`
(C) `temperature > 45`
(D) `temperature <= 60`
(E) Never correct

18. After discovering that the method did not work correctly, it was rewritten as follows.



```
public String chooseActivity(int temperature)
{
   String activity;
   if (temperature > 45)
      activity = "go horseback riding";
   else if (temperature > 60)
      activity = "go hiking";
   else if (temperature > 75)
      activity = "go swimming";
   else
      activity = "play ping pong";
   return activity;
}
```

Consider the following statement.



```
System.out.println("We recommend that you " + chooseActivity(temperature));
```

What is the largest temperature range for which the correct suggestion is printed (as defined above)?

(A) Always correct

(B) Never correct

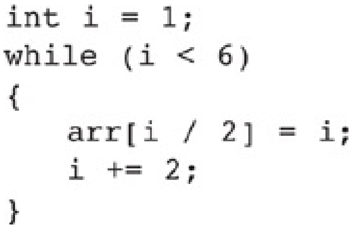
(C) `temperature <= 75`

(D) `temperature <= 60`

(E) `temperature <= 45`

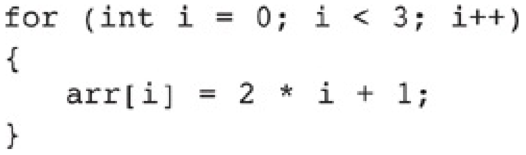
19. Assume `int[] arr` has been correctly instantiated and is of sufficient size. Which of these code segments results in identical arrays?

I.



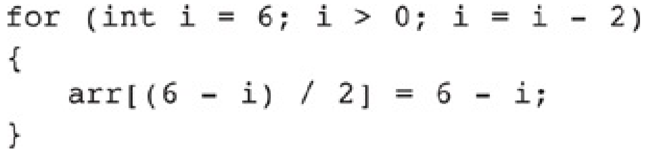
```
int i = 1;
while (i < 6)
{
   arr[i / 2] = i;
   i += 2;
}
```

II.



```
for (int i = 0; i < 3; i++)
{
   arr[i] = 2 * i + 1;
}
```

III.



```
for (int i = 6; i > 0; i = i - 2)
{
   arr[(6 - i) / 2] = 6 - i;
}
```

(A) I and II only

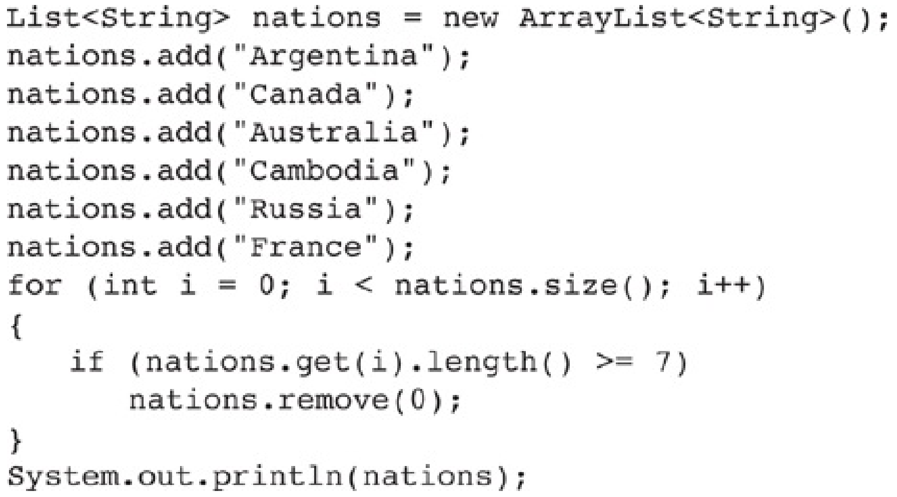
(B) II and III only

(C) I and III only

(D) I, II, and III

(E) All three outputs are different.

20. Consider the following code segment.



```
List<String> nations = new ArrayList<String>();
nations.add("Argentina");
nations.add("Canada");
nations.add("Australia");
nations.add("Cambodia");
nations.add("Russia");
nations.add("France");
for (int i = 0; i < nations.size(); i++)
{
   if (nations.get(i).length() >= 7)
      nations.remove(0);
}
System.out.println(nations);
```

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

(A) `[Canada, Russia, France]`

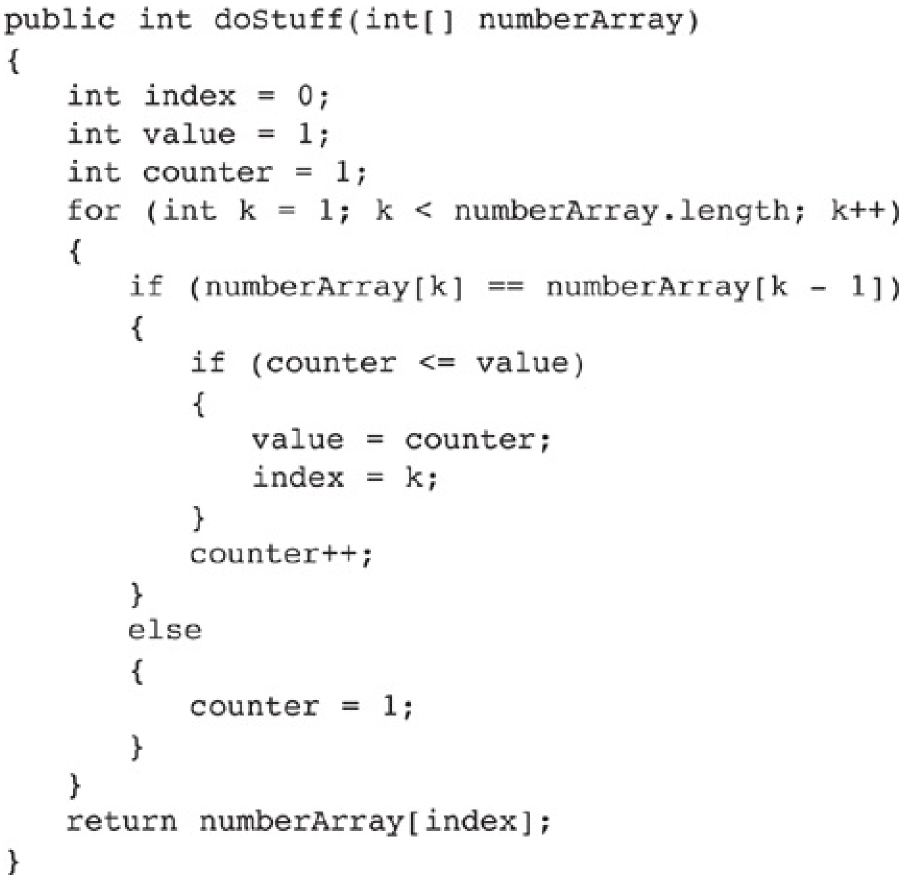
(B) `[Cambodia, Russia, France]`

(C) `[Australia, Cambodia, Russia, France]`

(D) `[Canada, Cambodia, Russia, France]`

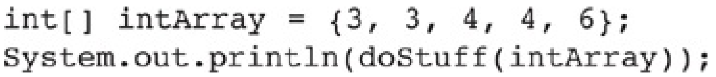
(E) Nothing is printed. `IndexOutOfBoundsException`

21. Consider the following method.



```
public int doStuff(int[] numberArray)
{
   int index = 0;
   int value = 1;
   int counter = 1;
   for (int k = 1; k < numberArray.length; k++)
   {
      if (numberArray[k] == numberArray[k - 1])
      {
         if (counter <= value)
         {
            value = counter;
            index = k;
         }
         counter++;
      }
      else
      {
         counter = 1;
      }
   }
   return numberArray[index];
}
```

Consider the following code segment.



```
int[] intArray = {3, 3, 4, 4, 6};
System.out.println(doStuff(intArray));
```

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

(A) `1`

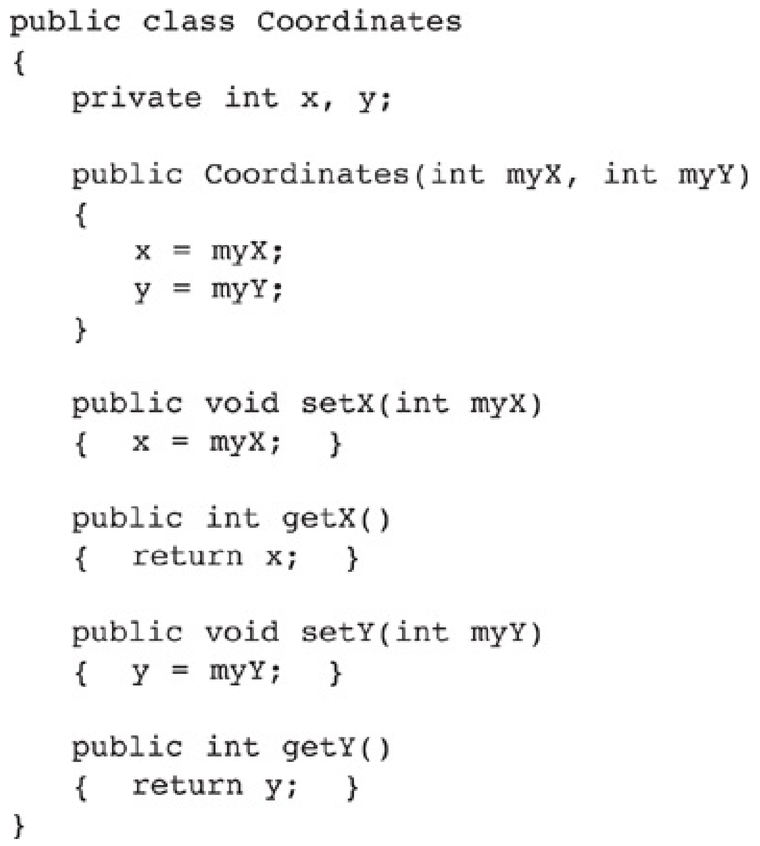
(B) `3`

(C) `4`

(D) `5`

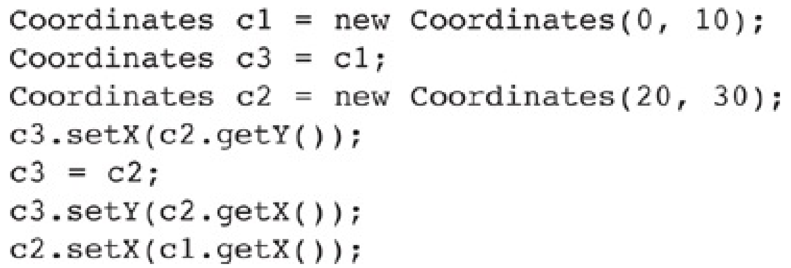
(E) `6`

Questions 22–23 refer to the following class.



```
public class Coordinates
{
   private int x, y;

   public Coordinates(int myX, int myY)
   {
      x = myX;
      y = myY;
   }

   public void setX(int myX)
   {  x = myX;   }

   public int getX()
   {  return x;   }

   public void setY(int myY)
   {  y = myY;   }

   public int getY()
   {  return y;   }
}
```

Consider the following code segment.



```
Coordinates c1 = new Coordinates(0, 10);
Coordinates c3 = c1;
Coordinates c2 = new Coordinates(20, 30);
c3.setX(c2.getY());
c3 = c2;
c3.setY(c2.getX());
c2.setX(c1.getX());
```

22. Which of the following calls returns the value 20?

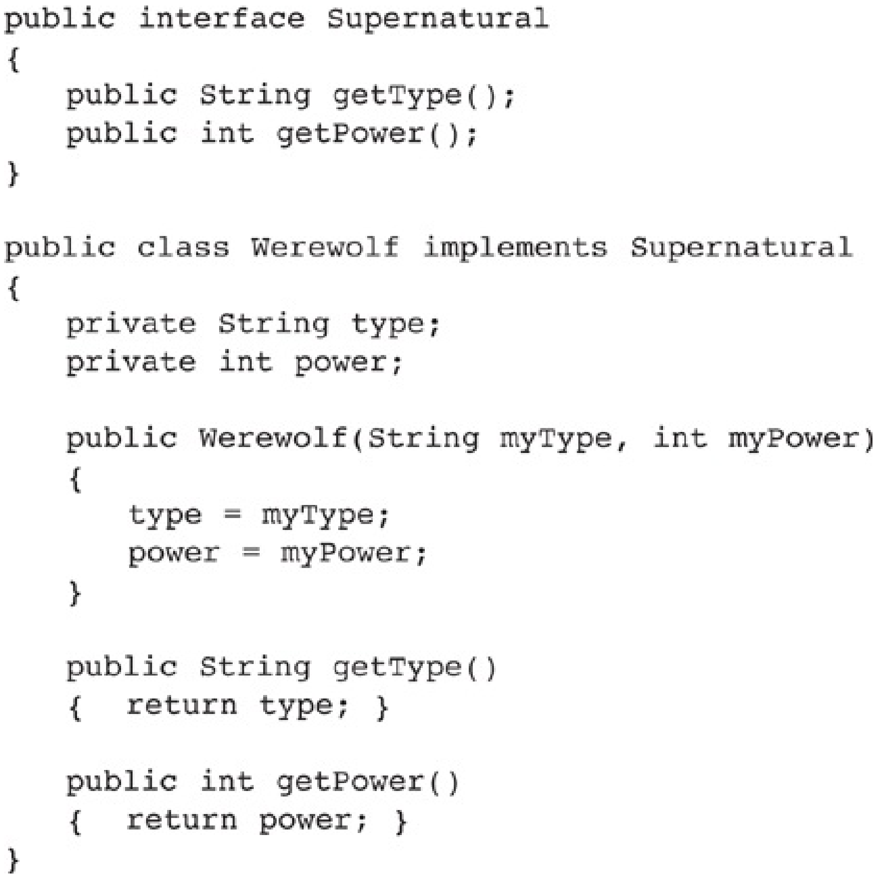
(A) `c1.getX()`
(B) `c1.getY()`
(C) `c2.getX()`
(D) `c2.getY()`
(E) `c3.getX()`

23. Which of the following calls returns the value 30?

(A) `c1.getX()`
(B) `c2.getX()`
(C) `c3.getX()`
(D) All of the above

(E) None of the above

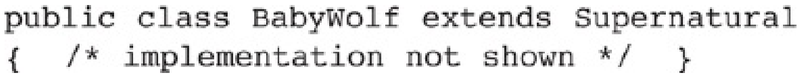
Questions 24–25 refer to the following interface and class.



```
public interface Supernatural
{
   public String getType();
   public int getPower();
}

public class Werewolf implements Supernatural
{
   private String type;
   private int power;

   public Werewolf(String myType, int myPower)
   {
      type = myType;
      power = myPower;
   }

   public String getType()
   {  return type; }

   public int getPower()
   {  return power; }
}
```

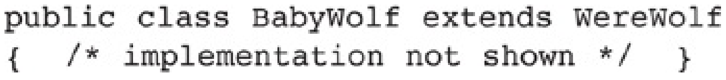
24. Consider the following class declarations.

I.



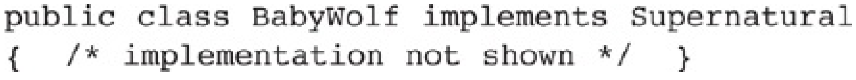
```
public class BabyWolf extends Supernatural
{  /* implementation not shown */  }
```

II.



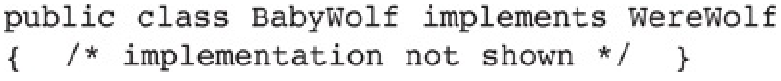
```
public class BabyWolf extends WereWolf
{  /* implementation not shown */  }
```

III.



```
public class BabyWolf implements Supernatural
{  /* implementation not shown */  }
```

IV.



```
public class BabyWolf implements WereWolf
{  /* implementation not shown */  }
```

Which of the declarations does not cause an error?

(A) I and II only

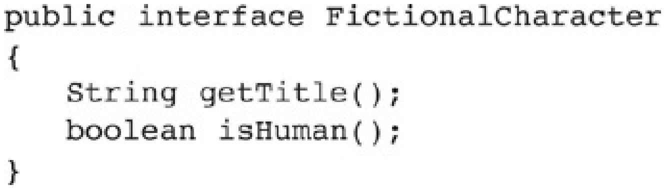
(B) II and III only

(C) III and IV only

(C) I and III only

(E) I and IV only

25. Consider the following interface.



```
public interface FictionalCharacter
{
   String getTitle();
   boolean isHuman();
}
```

Which of the following class declarations does not cause an error?

(A) `public class A extends Supernatural implements FictionalCharacter`

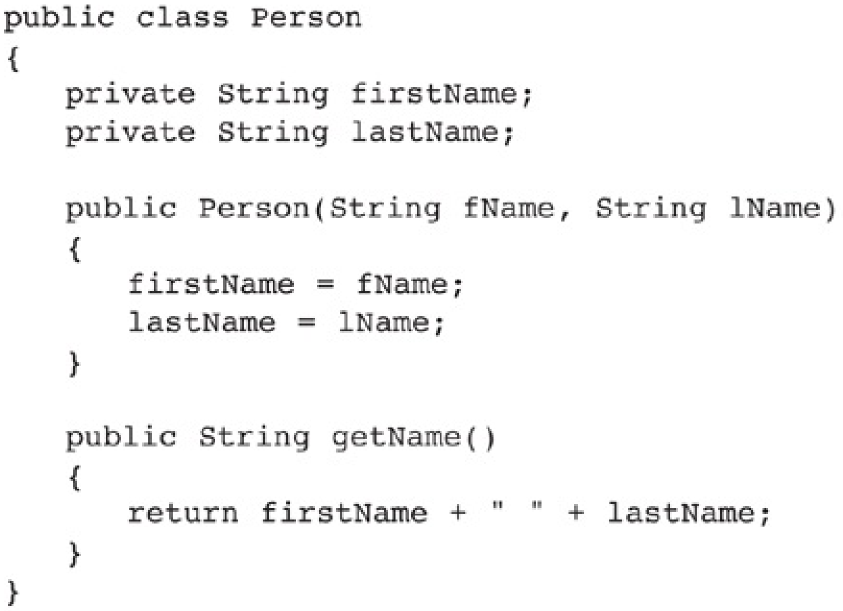
(B) `public class B extends BabyWolf, WereWolf implements FictionalCharacter`

(C) `public class C implements Supernatural, Werewolf`

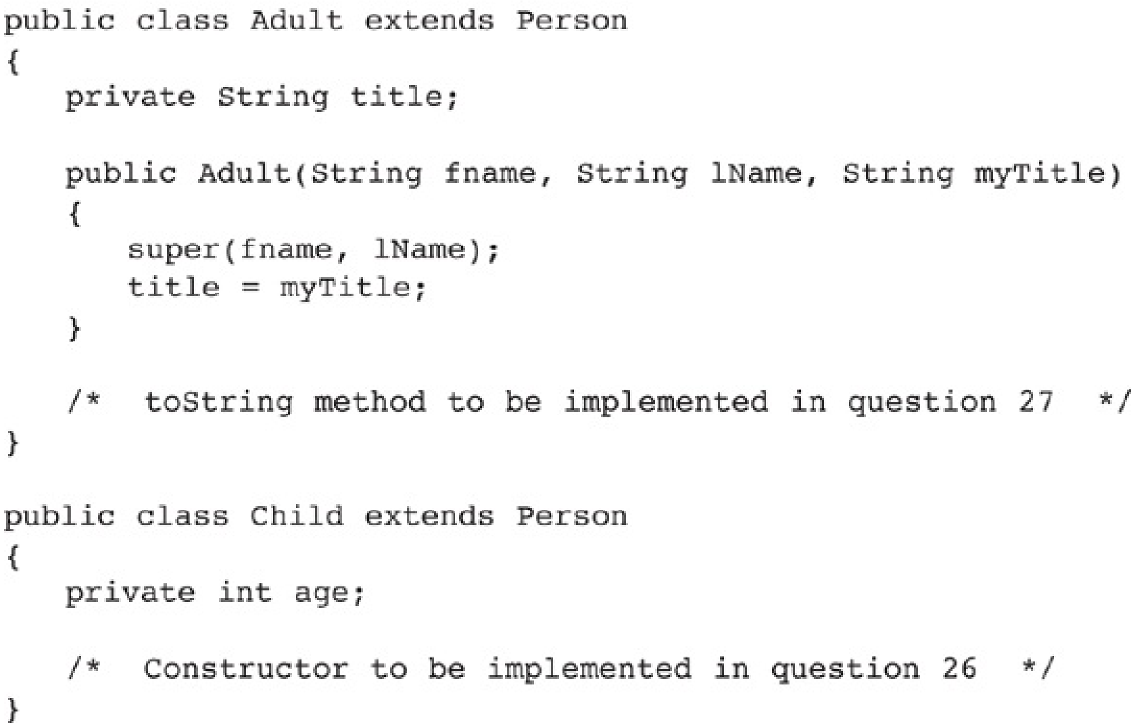
(D) `public class E extends FictionalCharacter`

(E) `public class D extends Werewolf implements Supernatural, FictionalCharacter`

Questions 26–28 refer to the following classes.



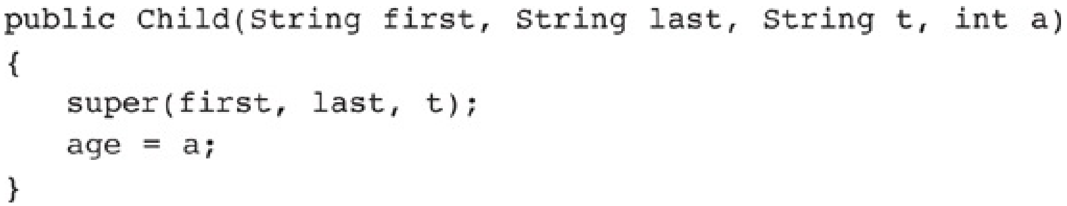
```
public class Person
{
   private String firstName;
   private String lastName;

   public Person(String fName, String lName)
   {
      firstName = fName;
      lastName = lName;
   }

   public String getName()
   {
      return firstName + " " + lastName;
   }
}
```



```
public class Adult extends Person
{
   private String title;

   public Adult(String fname, String lName, String myTitle)
   {
      super(fname, lName);
      title = myTitle;
   }

   /*  toString method to be implemented in question 27   */
}

public class Child extends Person
{
   private int age;

   /*  Constructor to be implemented in question 26   */
}
```

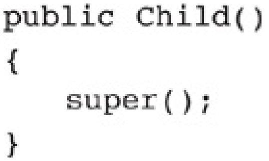
26. Which of the following is a correct implementation of a `Child` class constructor?

(A)



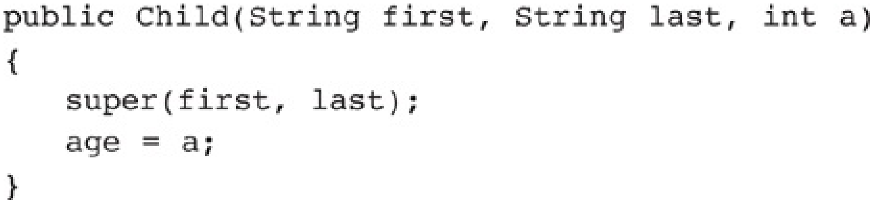
```
public Child(String first, String last, String t, int a)
{
   super(first, last, t);
   age = a;
}
```

(B)



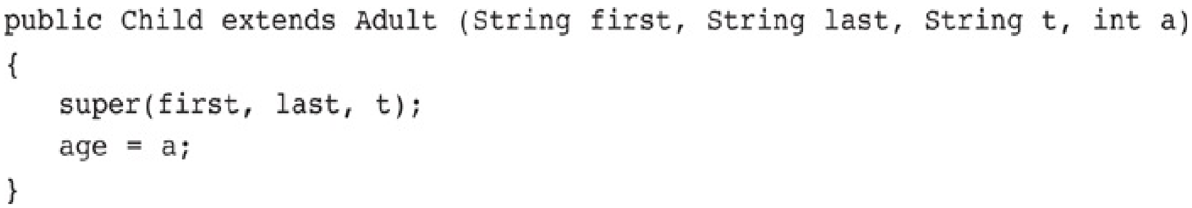
```
public Child()
{
   super();
}
```

(C)



```
public Child(String first, String last, int a)
{
   super(first, last);
   age = a;
}
```

(D)



```
public Child extends Adult (String first, String last, String t, int a)
{
   super(first, last, t);
   age = a;
}
```

(E)
```
public Child extends Person(String first, String last, int a)
{
   super(first, last);
   age = a;
}
```

27. Which of the following is a correct implementation of the `Adult` class `toString` method?

(A)
```
public String toString()
{
   System.out.println(title + " " + super.toString());
}
```

(B)
```
public String toString()
{
   return title + " " + super.toString();
}
```

(C)
```
public String toString()
{
   return title + " " + firstName + " " + lastName;
}
```

(D)
```
public void toString()
{
   System.out.println(title + " " + super.getName());
}
```

(E)
```
public String toString()
{
   return title + " " + super.getName();
}
```

28. Consider the following method declaration in a different class.

```
public void findSomeone(Adult someone)
```

Assume the following variables have been correctly instantiated and initialized with appropriate values.

```
Person p;
Adult a;
Child c;
```

Which of the following method calls compiles without error?

I. `findSomeone(p);`

II. `findSomeone(a);`

III. `findSomeone(c);`

IV. `findSomeone((Adult) p);`

V. `findSomeone((Adult) c);`

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

29. Consider the following method. The method is intended to return `true` if the value `val` raised to the power `power` is within the tolerance `tolerance` of the target value `target`, and `false` otherwise.

```
public boolean similar(double val, double power, double target,
double tolerance)
{
   /* missing code */
}
```

Which code segment below can replace */* missing code */* to make the method work as intended?

(A)
```
double answer = Math.pow(val, power);
if (answer - tolerance >= target)
   return true;
return false;
```

(B) `return (Math.abs(Math.pow(val, power))) - target <= tolerance;`

(C)
```
double answer = Math.pow(Math.abs(val), power);
return answer - target <= tolerance;
```

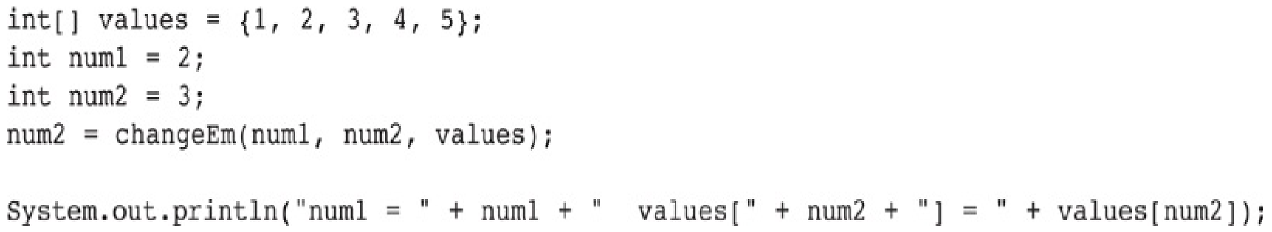
(D)
```
double answer = Math.pow(val, power) - target;
boolean within = answer - tolerance >= 0;
return within;
```

(E) `return (Math.abs(Math.pow(val, power) - target) <= tolerance);`

30. Consider the following method.

```
public int changeEm(int num1, int num2, int[] values)
{
   num1 = values[num2];
   values[num1] = num2;
   num2++;
   return num2;
}
```

Consider the following code segment.



```
int[] values = {1, 2, 3, 4, 5};
int num1 = 2;
int num2 = 3;
num2 = changeEm(num1, num2, values);

System.out.println("num1 = " + num1 + "  values[" + num2 + "] = " + values[num2]);
```

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

(A) `num1 = 2  values[4] = 4`

(B) `num1 = 2  values[4] = 5`

(C) `num1 = 2  values[4] = 3`

(D) `num1 = 4  values[3] = 3`

(E) `num1 = 4  values[3] = 4`

Questions 31–32 refer to the following information.

In computer graphics, RGB colors can be specified several ways.

- Colors can be represented by ordered triples in which the first number represents red, the second green, and the third blue. The three numbers range in value from 0 to 255 (inclusive).
- Colors can be represented by a single six-digit hexadecimal number written with a # in front of it.

These two representations have a simple relationship. The leftmost two digits of the hexadecimal number are a direct conversion from decimal to hexadecimal of the red value in the ordered triple, the middle two digits are a direct conversion of the green value in the ordered triple, and the rightmost two digits are a direct conversion of the blue value in the ordered triple.

For example, the ordered triple (10, 16, 20) becomes #0A1014 in hexadecimal form since 10→0A, 16→ 10, and 20 → 14, and the hexadecimal #110F03 becomes the ordered triple (17, 15, 3).

31. Convert the RGB ordered triple (125, 200, 78) to hexadecimal form.

(A) 12520078

(B) 7DC84E

(C) 713128414

(D) 202018

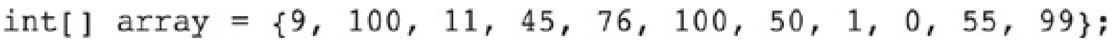
(E) 84D452

32. Convert the RGB color #EA27B5 in hexadecimal form to an ordered triple.

(A) (1410, 27, 115)

(B) (204, 40, 115)
(C) (234, 39, 181)
(D) (221, 39, 171)
(E) (24, 39, 16)

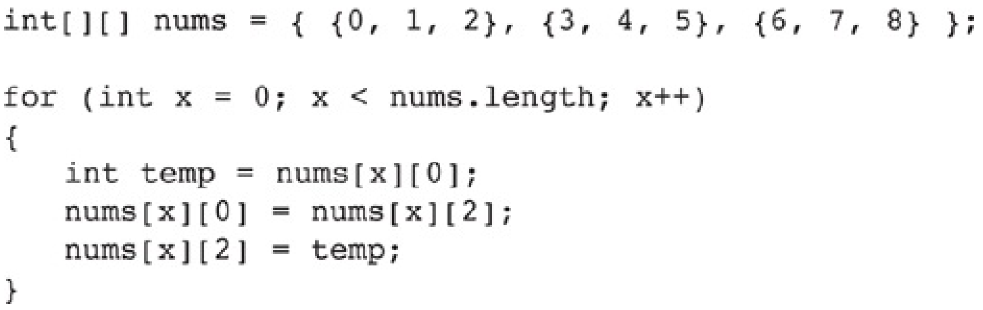
33. A programmer intends to apply the standard Binary Search algorithm on the following array of integers. The standard Binary Search algorithm returns the index of the search target if it is found and -1 if the target is not found. What is returned by the algorithm when a search for 50 is executed?



```
int[] array = {9, 100, 11, 45, 76, 100, 50, 1, 0, 55, 99};
```

(A) `-1`
(B) `0`
(C) `5`
(D) `6`
(E) `7`

34. Consider the following code segment.

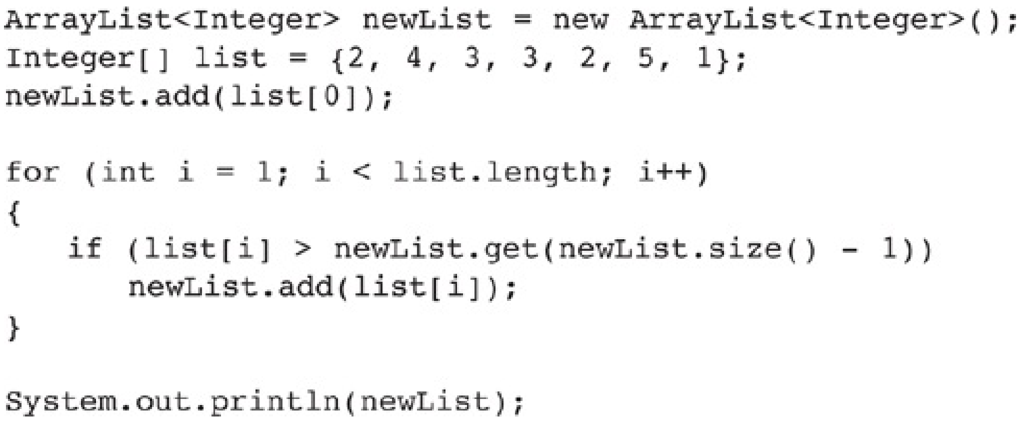


```
int[][] nums = { {0, 1, 2}, {3, 4, 5}, {6, 7, 8} };

for (int x = 0; x < nums.length; x++)
{
    int temp = nums[x][0];
    nums[x][0] = nums[x][2];
    nums[x][2] = temp;
}
```

What are the values in array nums after the code segment is executed?

(A) `{ {0, 1, 0}, {3, 4, 3}, {6, 7, 6} }`
(B) `{ {0, 0, 0}, {3, 3, 3}, {6, 6, 6} }`
(C) `{ {2, 1, 0}, {5, 4, 3}, {8, 7, 6} }`
(D) `{ {0, 1, 2}, {3, 4, 5}, {6, 7, 8} }`
(E) `{ {8, 7, 6}, {5, 4, 3}, {2, 1, 0} }`

35. Consider the following code segment.



```
ArrayList<Integer> newList = new ArrayList<Integer>();
Integer[] list = {2, 4, 3, 3, 2, 5, 1};
newList.add(list[0]);

for (int i = 1; i < list.length; i++)
{
   if (list[i] > newList.get(newList.size() - 1))
      newList.add(list[i]);
}

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

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

(A) `[2, 4, 5]`

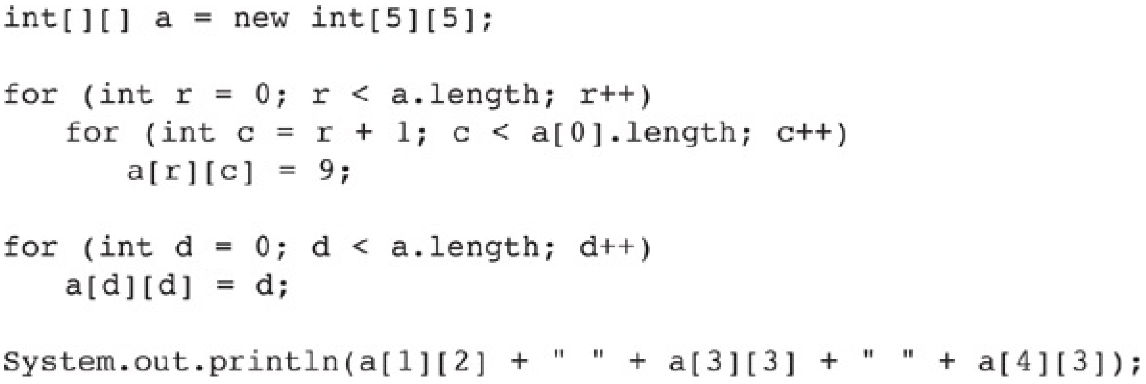
(B) `[2, 2, 1]`

(C) `[2, 4, 3, 5, 1]`

(D) `[1, 2, 3, 3, 4, 5]`

(E) `[2, 4, 3, 3, 5, 1]`

36. Consider the following code segment.



```
int[][] a = new int[5][5];

for (int r = 0; r < a.length; r++)
   for (int c = r + 1; c < a[0].length; c++)
      a[r][c] = 9;

for (int d = 0; d < a.length; d++)
   a[d][d] = d;

System.out.println(a[1][2] + " " + a[3][3] + " " + a[4][3]);
```

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

(A) `9 9 9`

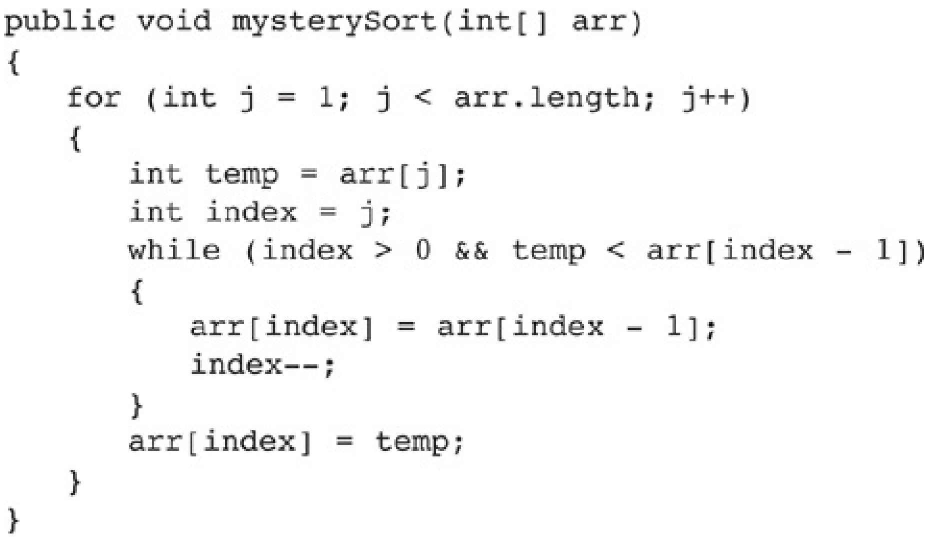
(B) `0 0 0`

(C) `0 3 9`

(D) `9 3 9`

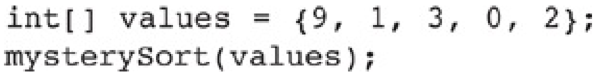
(E) `9 3 0`

37. Consider the following sorting method.



```
public void mysterySort(int[] arr)
{
   for (int j = 1; j < arr.length; j++)
   {
      int temp = arr[j];
      int index = j;
      while (index > 0 && temp < arr[index - 1])
      {
         arr[index] = arr[index - 1];
         index--;
      }
      arr[index] = temp;
   }
}
```

Consider the following code segment.



```
int[] values = {9, 1, 3, 0, 2};
mysterySort(values);
```

Which of the following shows the elements of the array in the correct order after the second pass through the outer loop of the sorting algorithm?

(A) `{0, 1, 2, 9, 3}`

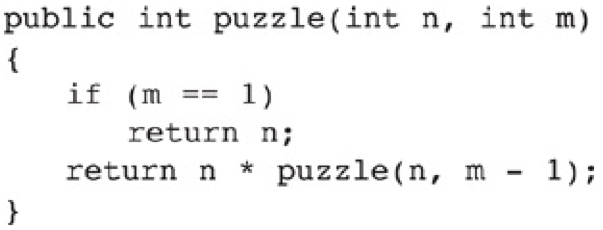
(B) `{0, 1, 3, 9, 2}`

(C) `{0, 1, 9, 3, 2}`

(D) `{1, 3, 9, 0, 2}`

(E) `{1, 3, 0, 2, 9}`

38. Consider the following method.



```
public int puzzle(int n, int m)
{
   if (m == 1)
      return n;
   return n * puzzle(n, m - 1);
}
```

What is returned by the method call `puzzle(3, 4)`?

(A) `9`

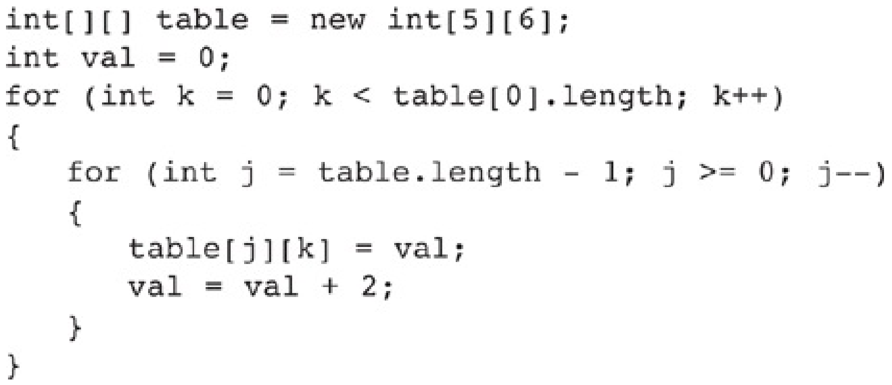
(B) `64`

(C) `81`

(D) `128`

(E) Nothing is returned. Infinite recursion causes a stack overflow error.

39. Consider the following code segment.



```
int[][] table = new int[5][6];
int val = 0;
for (int k = 0; k < table[0].length; k++)
{
   for (int j = table.length - 1; j >= 0; j--)
   {
      table[j][k] = val;
      val = val + 2;
   }
}
```

What is the value of `table[3][4]` after executing the code segment?

(A) `22`

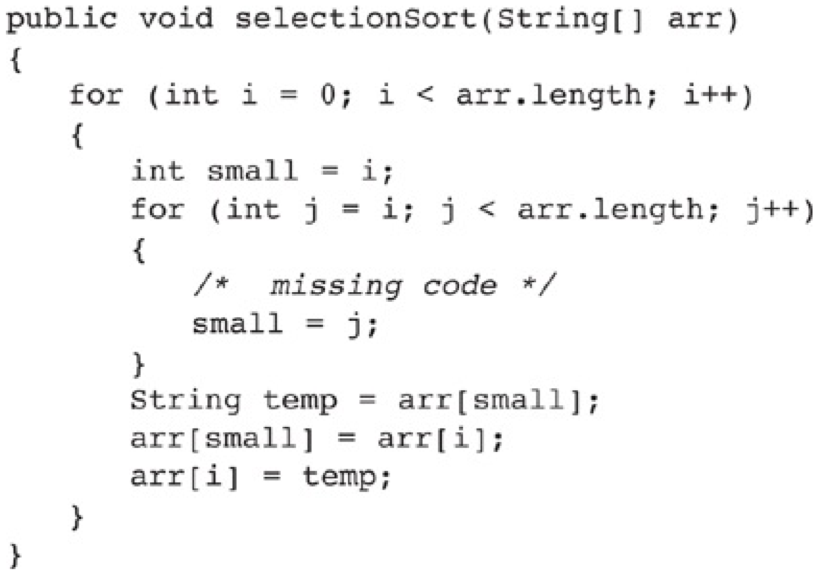
(B) `30`

(C) `34`

(D) `42`

(E) `46`

40. Consider the following method.



```
public void selectionSort(String[] arr)
{
   for (int i = 0; i < arr.length; i++)
   {
      int small = i;
      for (int j = i; j < arr.length; j++)
      {
         /*  missing code */
         small = j;
      }
      String temp = arr[small];
      arr[small] = arr[i];
      arr[i] = temp;
   }
}
```

Assume `String[] ray` has been properly instantiated and populated with `String` objects.

Which line of code should replace `/* missing code */` so that the method call `selectionSort(ray)` results in an array whose elements are sorted from least to greatest?

(A) `if (arr[j].equals(arr[small]))`

(B) `if (arr[j] > arr[small])`

(C) `if (arr[j] < arr[small])`

(D) `if (arr[j].compareTo(arr[small]) > 0)`

(E) `if (arr[j].compareTo(arr[small]) < 0)`

STOP. End of Part I.