**Review - MCQ**

**Review 6**

**Array**

1、Consider the following method.

public int[] transform(int[] a)

{

a[0]++;

a[2]++;

return a;

}

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

/\* missing code \*/

arr = transform(arr);

After executing the code segment, the array arr should contain {1, 0, 1, 0}. Which of the following can be used to replace /\* *missing code* \*/ so that the code segment works as intended?

I. int[] arr = {0, 0, 0, 0};

II. int[] arr = new int[0];

III. int[] arr = new int[4];

(A) I only

(B) II only

(C) III only

(D) I and II

(E) I and III

2、Consider the following code segment.

int[] arr = {10, 20, 30, 40, 50};

for(int x = 1; x < arr.length - 1; x++)

{

arr[x + 1] = arr[x] + arr[x + 1];

}

Which of the following represents the contents of arr after the code segment has been executed?

(A) {10, 20, 30, 70, 120}

(B) {10, 20, 50, 90, 50}

(C) {10, 20, 50, 90, 140}

(D) {10, 30, 60, 100, 50}

(E) {10, 30, 60, 100, 150}

**Array-For loop & for each loop**

1、Consider the following two code segments.

**I.**

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

for (int x = 0; x < arr.length; x++)

{

System.out.print(arr[x + 3]);

}

**II.**

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

for (int x : arr)

{

System.out.print(x + 3);

}

Which of the following best describes the behavior of code segment I and code segment II ?

(A) Both code segment I and code segment II will print 45.

(B) Both code segment I and code segment II will print 45678.

(C) Code segment I will cause an ArrayIndexOutOfBoundsException and code segment II will

print 45.

(D) Code segment I will cause an ArrayIndexOutOfBoundsException and code segment II will

print 45678.

(E) Both code segment I and code segment II will cause an ArrayIndexOutOfBoundsException.

**Reverse array**

1、In the code segment below, assume that the int array numArr has been properly declared and initialized. The code segment is intended to reverse the order of the elements in numArr. For example, if numArr initially contains {1, 3, 5, 7, 9}, it should contain {9, 7, 5, 3, 1} after the code segment executes.

/\* missing loop header \*/

{

int temp = numArr[k];

numArr[k] = numArr[numArr.length - k - 1];

numArr[numArr.length - k - 1] = temp;

}

Which of the following can be used to replace /\* *missing loop header* \*/ so that the code segment works as intended?

(A) for (int k = 0; k < numArr.length / 2; k++)

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

(C) for (int k = 0; k < numArr.length / 2; k--)

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

(E) for (int k = numArr.length - 1; k >= 0; k++)

**Found maximum value**

1、Consider the following code segment, which is intended to print the maximum value in an integer array values.

Assume that the array has been initialized properly and that it contains at least one element.

int maximum = /\* missing initial value \*/;

for (int k = 1; k < values.length; k++)

{

if (values[k] > maximum)

{

maximum = values[k];

}

}

System.out.println(maximum);

Which of the following should replace /\* *missing initial value* \*/ so that the code segment will work as intended?

(A) 0

(B) values[0]

(C) values[1]

(D) Integer.MIN\_VALUE

(E) Integer.MAX\_VALUE

**ArrayList method**

1、Consider the following code segment.

ArrayList<Integer> nums = new ArrayList<>();

nums.add(3);

nums.add(2);

nums.add(1);

nums.add(0);

nums.add(0, 4);

nums.set(3, 2);

nums.remove(3);

nums.add(2, 0);

Which of the following represents the contents of nums after the code segment has been executed?

(A) [2, 4, 3, 2, 0]

(B) [3, 2, 0, 1, 0]

(C) [4, 2, 0, 2, 0]

(D) [4, 3, 0, 2, 0]

(E) [4, 3, 0, 3, 0]

2、Consider the following code segment.

ArrayList<String> arrList = new ArrayList<String>();

arrList.add("A");

arrList.add("B");

arrList.add("C");

arrList.add("D");

for (int i = 0; i < arrList.size(); i++)

{

System.out.print(arrList.remove(i));

}

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

(A) AC

(B)

BD

(C) ABC

(D) ABCD

(E) Nothing is printed.

3、In the code segment below, myList is an ArrayList of integers. The code segment is intended to remove all elements with the value 0 from myList.

int j = 0;

while (j < myList.size())

{

if (myList.get(j) == 0)

{

myList.remove(j);

}

j++;

}

The code segment does not always work as intended. For which of the following lists does the code segment NOT

produce the correct result?

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

(B) {0, 1, 0, 2}

(C) {1, 0, 0, 2}

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

(E) {1, 2, 3, 4}

**2D array**

1、Consider the following code segment, which is intended to display "cat".

String[][] keyboard = {{"q", "w", "e", "r", "t"},

{"a", "s", "d", "f", "g"},

{"z", "x", "c", "v", "b"}};

System.out.println(/\* missing expression \*/);

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

(A) keyboard[12] + keyboard[5] + keyboard[4]

(B) keyboard[13] + keyboard[6] + keyboard[5]

(C) keyboard[2][2] + keyboard[1][0] + keyboard[0][4]

(D) keyboard[2][2] + keyboard[0][1] + keyboard[4][0]

(E) keyboard[3][3] + keyboard[2][1] + keyboard[1][5]

2、A two-dimensional array myArray is to be created with the following contents.

{{0, 0, 3},

{0, 0, 0},

{7, 0, 0}}

Which of the following code segments can be used to correctly create and initialize myArray ?

I. int myArray[][] = new int[3][3];

myArray[0][2] = 3;

myArray[2][0] = 7;

II. int myArray[][] = new int[3][3];

myArray[0][2] = 7;

myArray[2][0] = 3;

III. int myArray[][] = {{0, 0, 3}, {0, 0, 0}, {7, 0, 0}};

(A) I only

(B) II only

(C) III only

(D) I and III

(E) II and III

3、Consider the following code segment.

int[][] arr = {{1, 3, 4}, {4, 5, 3}};

int max = arr[0][0];

for (int row = 0; row < arr.length; row++)

{

for (int col = 0; col < arr[row].length; col++)

{

int temp = arr[row][col];

if (temp % 2 == 0)

{

arr[row][col] = temp + 1; // line 11

}

if (temp > max)

{

max = temp;

}

}

}

System.out.println(max);

How many times will the statement in line 11 be executed as a result of executing the code segment?

(A) 2

(B) 3

(C) 4

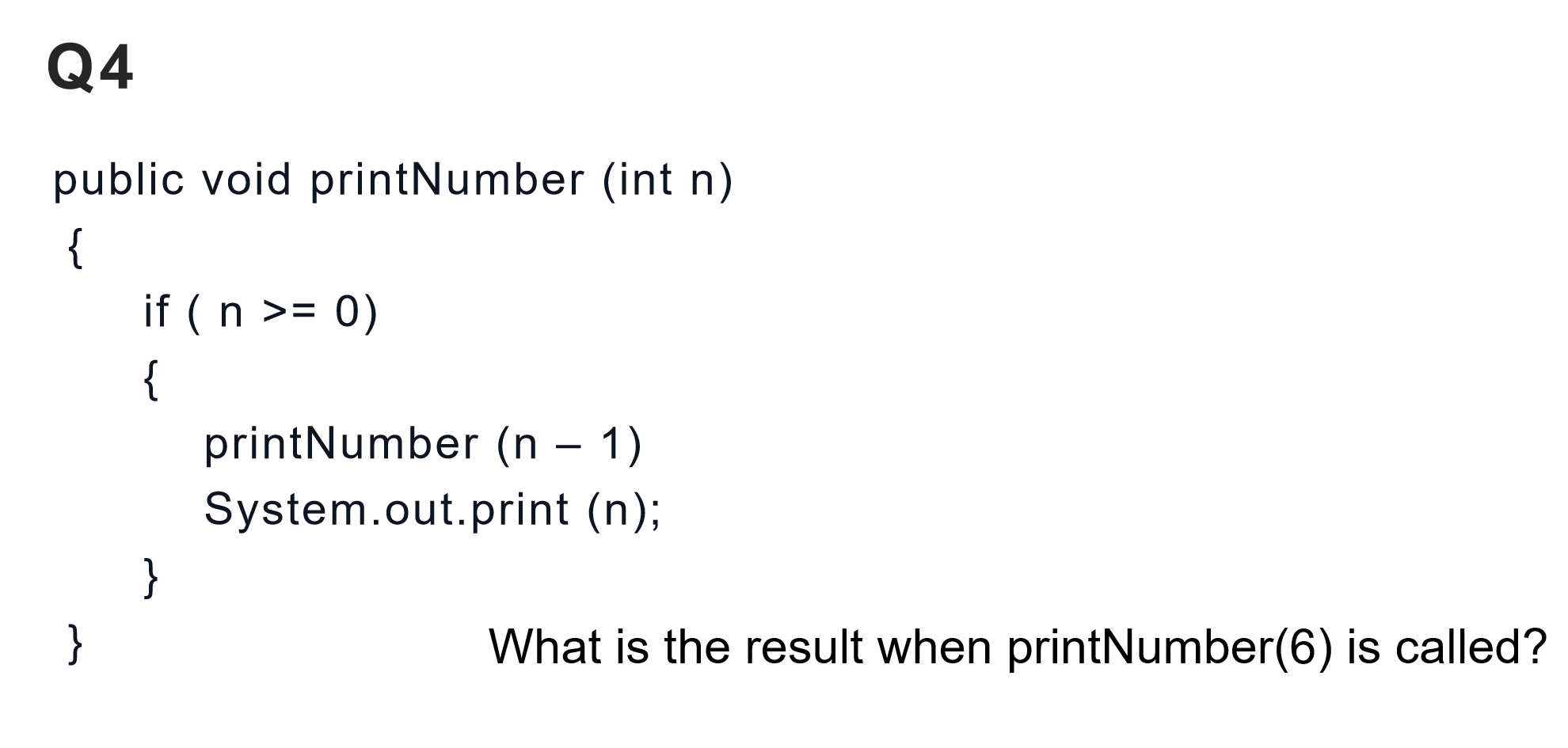
(D) 5

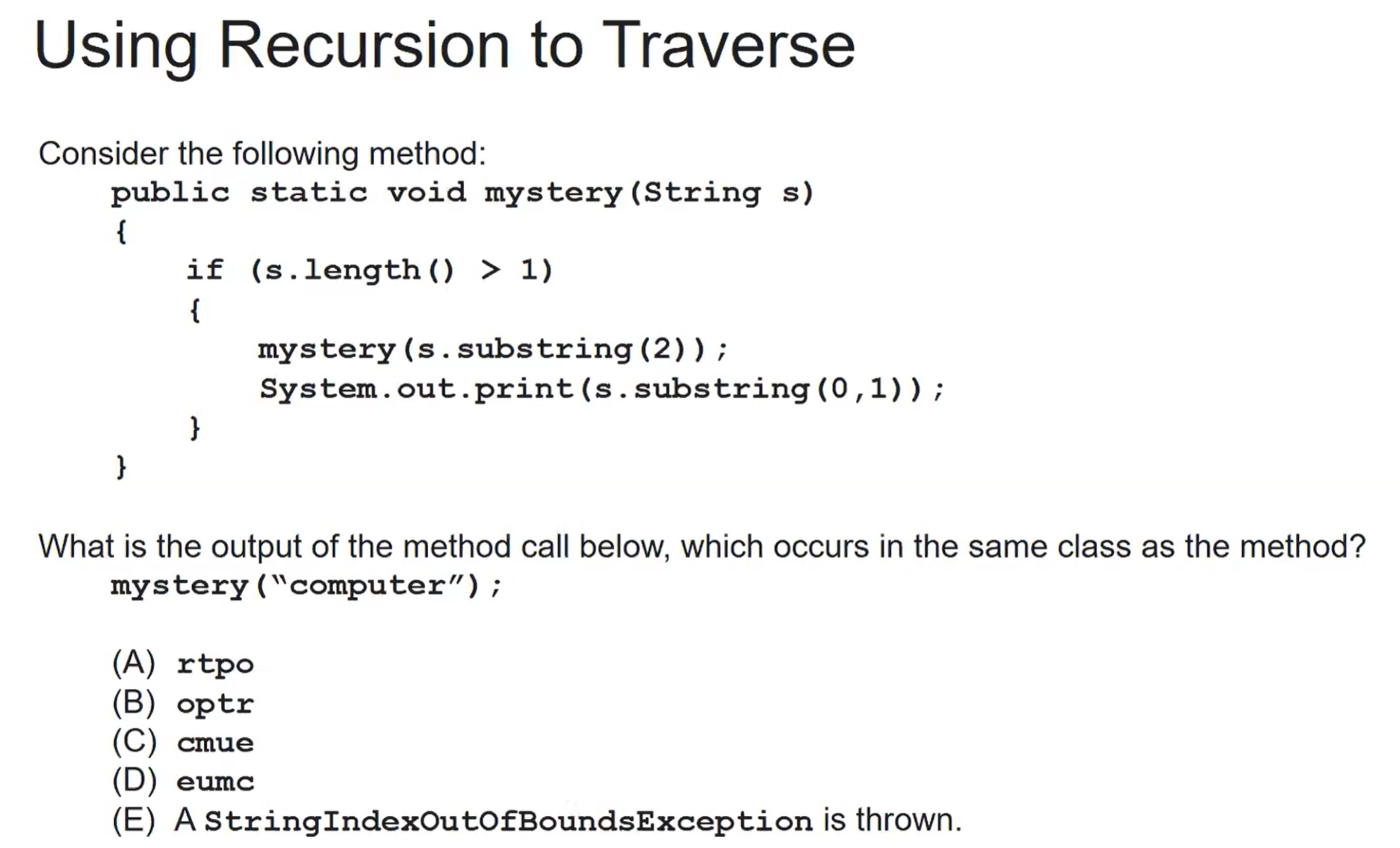
(E) 6

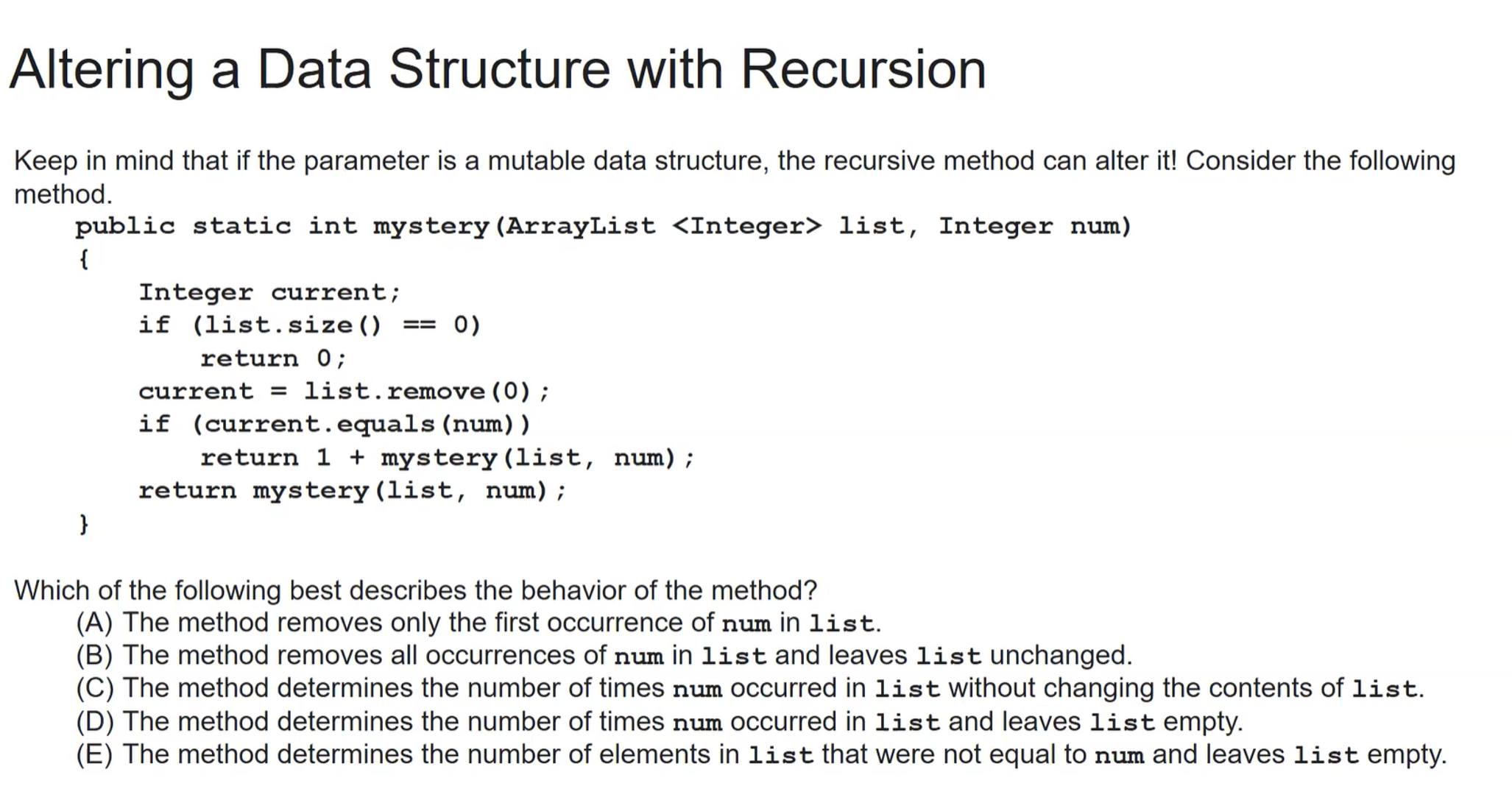
**Review 7**

**Searching & Sorting**

**Recursion**







**Binary search**

Quiz1

public int binarySearch(int[] elements, int target) {

int left = 0;

int right = elements.length - 1;

while (left <= right)

{

int middle = (left + right) / 2;

if (target < elements[middle])

{

right = middle - 1;

}

else if (target > elements[middle])

{

left = middle + 1;

}

else {

return middle;

}

}

return -1;

}

How many times would the loop in the binary search run for an array int[] arr = {2, 10, 23, 31, 55, 86} with binarySearch(arr,55)?

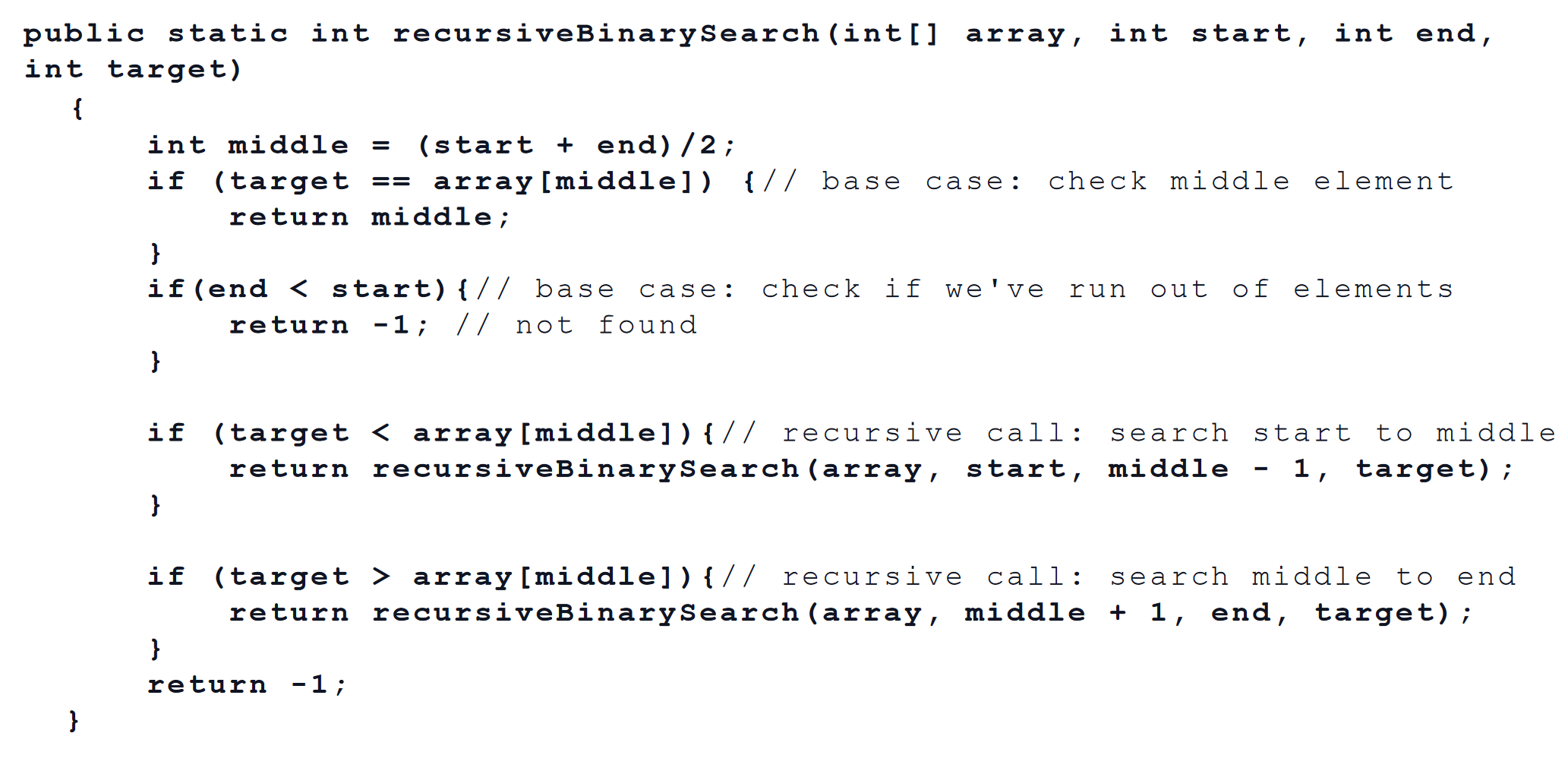
A. 2  
B. 1  
C. 3

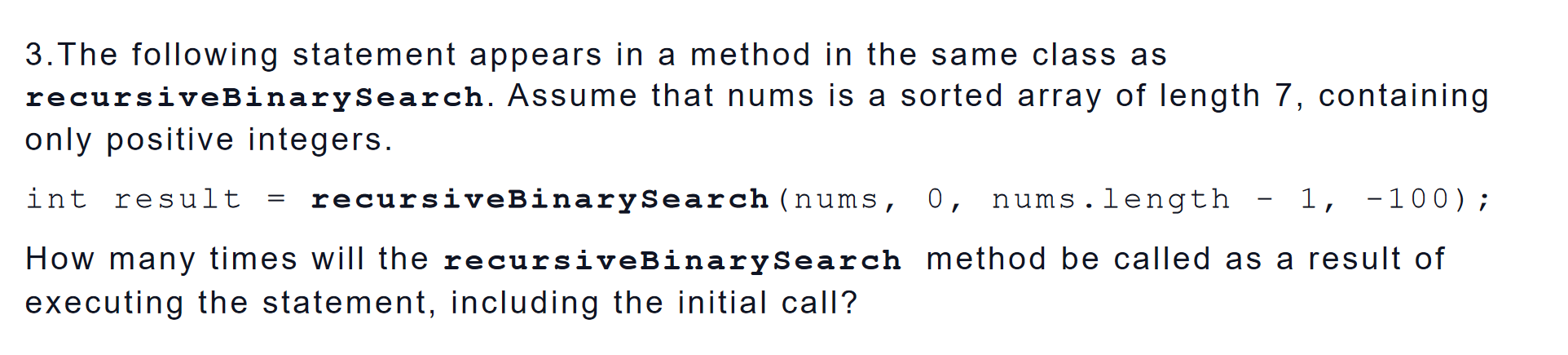
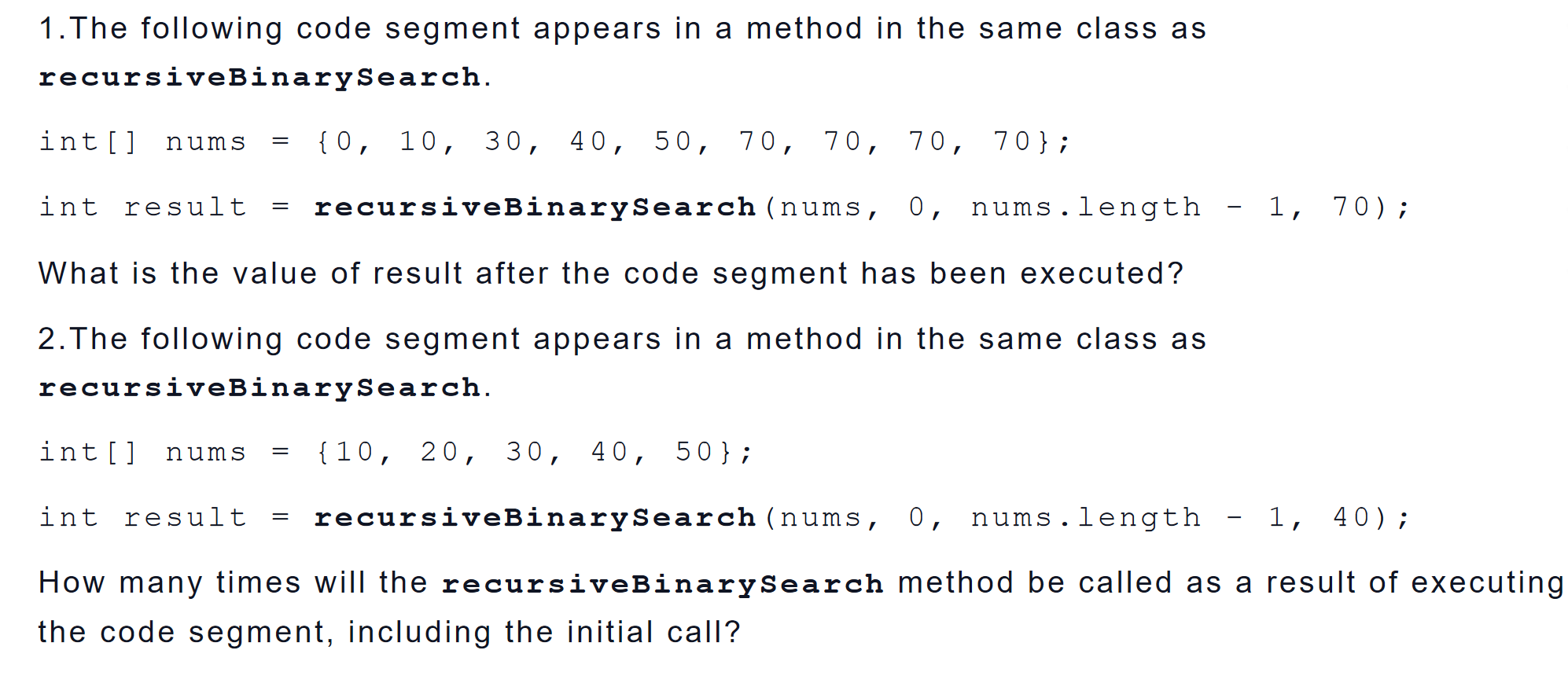
Quiz2

If you had an ordered array of size 500, what is the maximum number of iterations required to find an element with binary search?

1. approximately 15 times  
   B. approximately 9 times  
   C. 500 times  
   D. 2 times

**Binary search recursion**





**insertion sort**

1、Consider the following correct implementation of the insertion sort algorithm.

public static void insertionSort(int[] elements)

{

for (int j = 1; j < elements.length; j++)

{

int temp = elements[j];

int possibleIndex = j;

{

elements[possibleIndex] = elements[possibleIndex - 1];

possibleIndex--;

}

elements[possibleIndex] = temp; // line 12

}

}

The following declaration and method call appear in a method in the same class as insertionSort.

int[] nums = {8, 7, 5, 4, 2, 1};

insertionSort(nums);

How many times is the statement elements[possibleIndex] = temp; in line 12 of the method executed

as a result of the call to insertionSort ?

(A) 3

(B) 4

(C) 5

(D) 6

(E) 7

**Selection sort**

2、Consider the following correct implementation of the selection sort algorithm.

public static void selectionSort(int[] elements)

{

for (int j = 0; j < elements.length - 1; j++)

{

int minIndex = j;

for (int k = j + 1; k < elements.length; k++)

{

if (elements[k] < elements[minIndex])

{

minIndex = k;

}

}

if (j != minIndex)

{

int temp = elements[j];

elements[j] = elements[minIndex];

elements[minIndex] = temp; // line 19

}

}

}

The following declaration and method call appear in a method in the same class as selectionSort.

int[] arr = {30, 40, 10, 50, 20};

selectionSort(arr);

How many times is the statement elements[minIndex] = temp; in line 19 of the method executed as a result of the call to selectionSort ?

(A) 1

(B) 2

(C) 3

(D) 4

(E) 5

**Merge sort**

Consider the following code segment, which appears in a method in the same class as mergeSortHelper and merge.

int[] arr1 = {9, 1, 3, 5, 4};

int[] temp = new int[arr1.length];

mergeSortHelper(arr1, 0, arr1.length - 1, temp);

1.Which of the following represents the arrays merged the first time the merge method is executed as a result of the code segment above?

2.Which of the following represents the arrays merged the last time the merge method is executed as a result of the code segment above?

A {9} and {1} are merged to form {1, 9} .

B {1, 9} and {3} are merged to form {1, 3, 9} .

C {1, 9} and {5, 4} are merged to form {1, 4, 5, 9} .

D {1, 3, 9} and {5} are merged to form {1, 3, 5, 9} .

E {1, 3, 9} and {4, 5} are merged to form {1, 3, 4, 5, 9}.

3.How many times will the merge method be called as a result of executing the code segment?

A 1 B2 C3 D4 E5