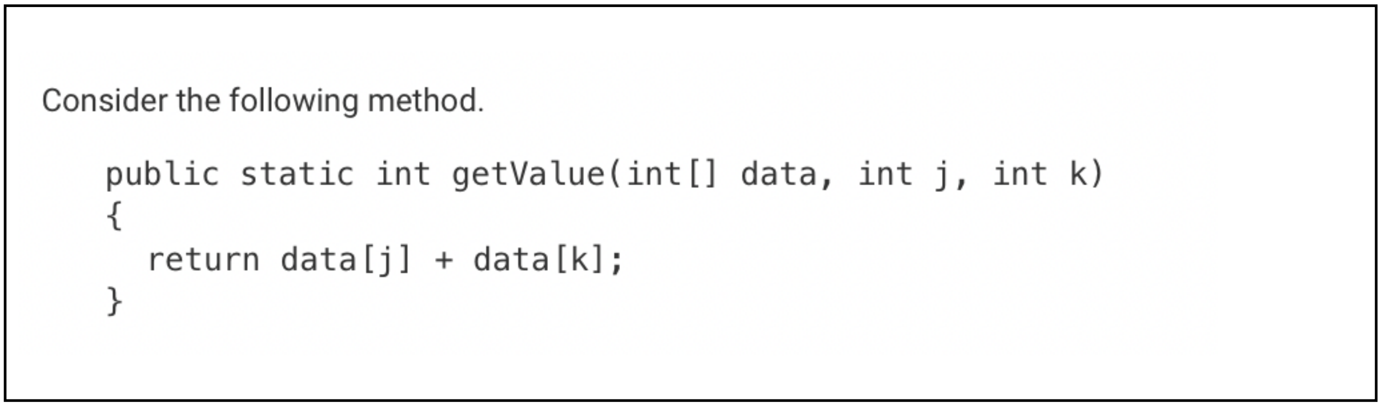
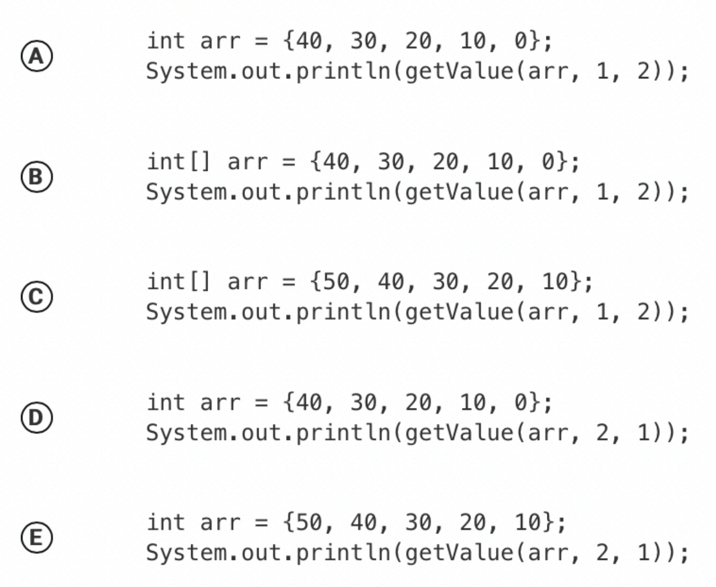
Problem #1



Consider the following method.

```
public static int getValue(int[] data, int j, int k)
{
  return data[j] + data[k];
}
```

Which of the following code segments, when appearing in another method in the same class as `getValue`, will print the value `70` ?



Ⓐ
```
int arr = {40, 30, 20, 10, 0};
System.out.println(getValue(arr, 1, 2));
```

Ⓑ
```
int[] arr = {40, 30, 20, 10, 0};
System.out.println(getValue(arr, 1, 2));
```

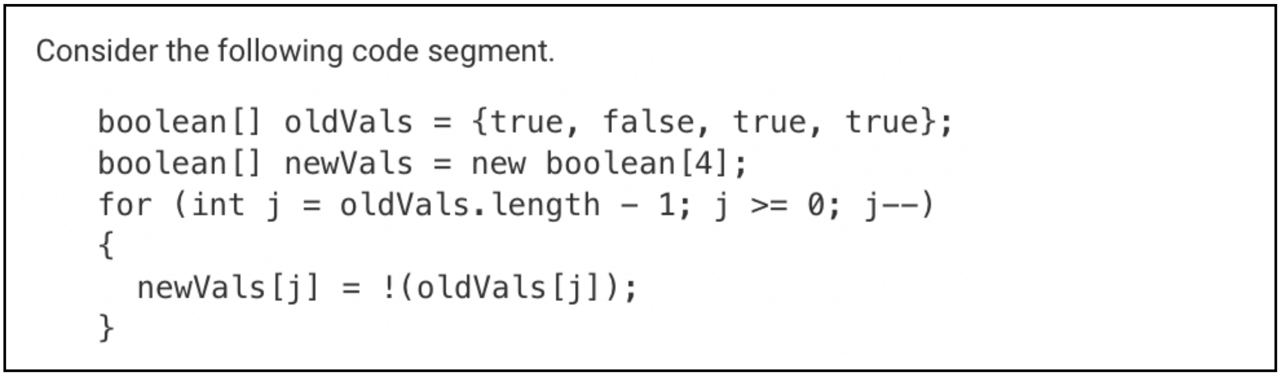
Ⓒ
```
int[] arr = {50, 40, 30, 20, 10};
System.out.println(getValue(arr, 1, 2));
```

Ⓓ
```
int arr = {40, 30, 20, 10, 0};
System.out.println(getValue(arr, 2, 1));
```

Ⓔ
```
int arr = {50, 40, 30, 20, 10};
System.out.println(getValue(arr, 2, 1));
```

Explain your answer (using a <u>code tracing table</u> or something else):

Problem #2



Consider the following code segment.

```
boolean[] oldVals = {true, false, true, true};
boolean[] newVals = new boolean[4];
for (int j = oldVals.length - 1; j >= 0; j--)
{
  newVals[j] = !(oldVals[j]);
}
```

What, if anything, will be the contents of `newVals` as a result of executing the code segment?

A. {true, true, false, true}

B. {true, false, true, true}

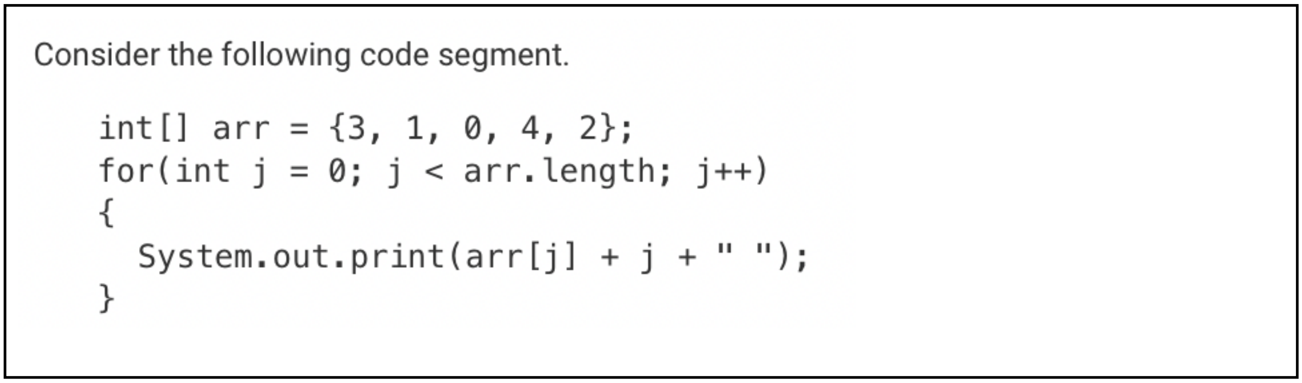
C. {false, true, false, false}

D. {false, false, true, false}

E. The array newVals will not contain any values because the code segment does not compile.

Explain your answer (using a <u>code tracing table</u> or something else):

Problem #3



Consider the following code segment.

```
int[] arr = {3, 1, 0, 4, 2};
for(int j = 0; j < arr.length; j++)
{
  System.out.print(arr[j] + j + " ");
}
```

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

A. 3 1 0 4 2

B. 3 2 2 7 6

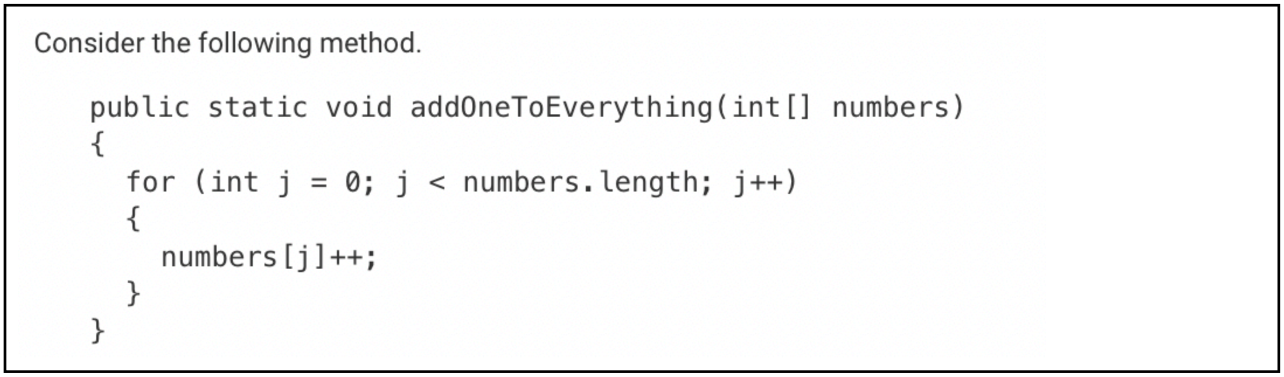
C. 6 2 0 8 4

D. 7 2 3 6 2

E. Nothing is printed, because an ArrayIndexOutOfBounds Exception is thrown.

Explain your answer (using a <u>code tracing table</u> or something else):

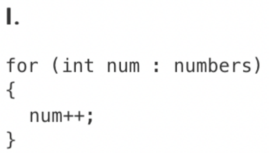
Problem #4



Consider the following method.

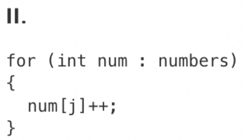
```
public static void addOneToEverything(int[] numbers)
{
  for (int j = 0; j < numbers.length; j++)
  {
    numbers[j]++;
  }
}
```

Which of the following code segments, if any, can be used to replace the body of the method so that `numbers` will contain the same values?



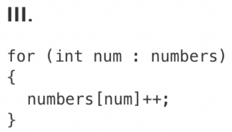
I.

```
for (int num : numbers)
{
  num++;
}
```



II.

```
for (int num : numbers)
{
  num[j]++;
}
```



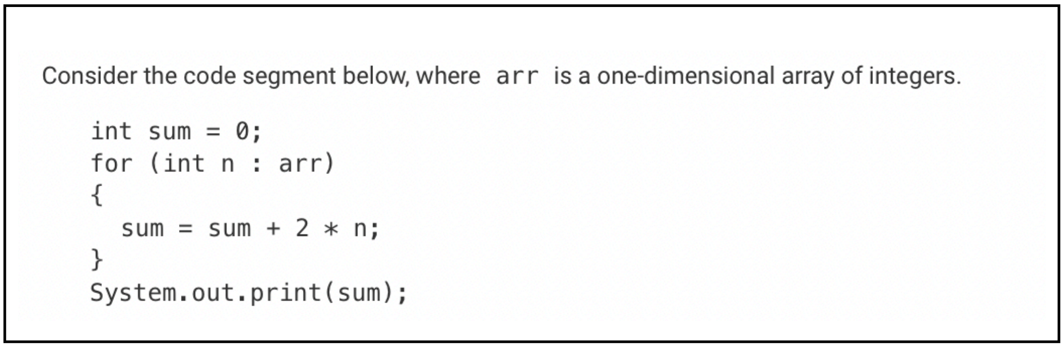
III.

```
for (int num : numbers)
{
  numbers[num]++;
}
```

A. I only
B. I and III only
C. II and III only
D. I, II, and III
E. None of the code segments will return an equivalent result.

Explain your answer (using a <u>code tracing table</u> or something else):

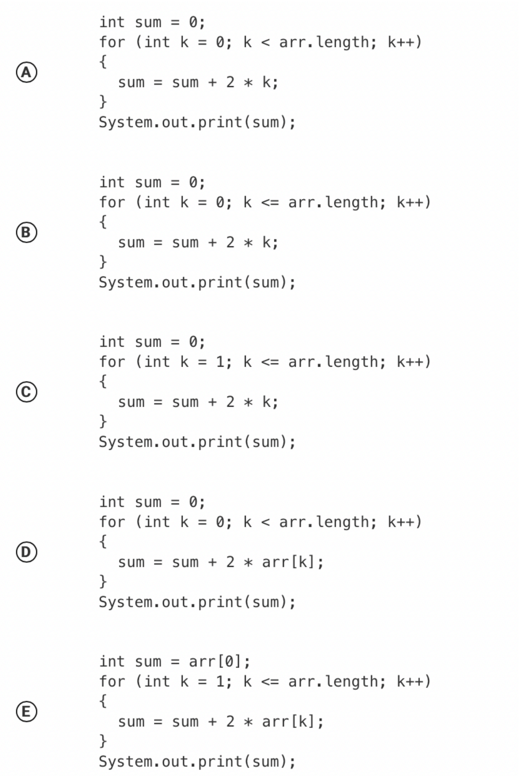
Problem #5



Consider the code segment below, where `arr` is a one-dimensional array of integers.

```
int sum = 0;
for (int n : arr)
{
  sum = sum + 2 * n;
}
System.out.print(sum);
```

Which of the following code segments will produce the same output as the code



Ⓐ

```
int sum = 0;
for (int k = 0; k < arr.length; k++)
{
  sum = sum + 2 * k;
}
System.out.print(sum);
```

Ⓑ

```
int sum = 0;
for (int k = 0; k <= arr.length; k++)
{
  sum = sum + 2 * k;
}
System.out.print(sum);
```

Ⓒ

```
int sum = 0;
for (int k = 1; k <= arr.length; k++)
{
  sum = sum + 2 * k;
}
System.out.print(sum);
```

Ⓓ

```
int sum = 0;
for (int k = 0; k < arr.length; k++)
{
  sum = sum + 2 * arr[k];
}
System.out.print(sum);
```

Ⓔ

```
int sum = arr[0];
for (int k = 1; k <= arr.length; k++)
{
  sum = sum + 2 * arr[k];
}
System.out.print(sum);
```

Explain your answer (using a code tracing table or something else):