Question 8

Consider the following method.

```
public static int[] operation(int[][] matrix, int r, int c)
{
  int[] result = new int[matrix.length];

  for (int j = 0 ; j < matrix.length ; j++)
  {
    result[j] = matrix[r][j] * matrix[j][c];
  }
  return result;
}
```

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

```
int[][] mat = {{3, 2, 1, 4},
               {1, 2, 3, 4},
               {2, 2, 1, 2},
               {1, 1, 1, 1}};

int[] arr = operation(mat, 1, 2);
```

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

My Answer: {2, 2, 4, 4}
Correct Answer: {1, 6, 3, 4}

Explanation:
int [] result=new[matrix.length]; // create result array length of 4, because matrix.length=4

r=1 and c=2

for(int j=0;j<matric.length;j++) // loop exicute 4 times{

result[j]=matrix[r][j]*matrix[j][c];}

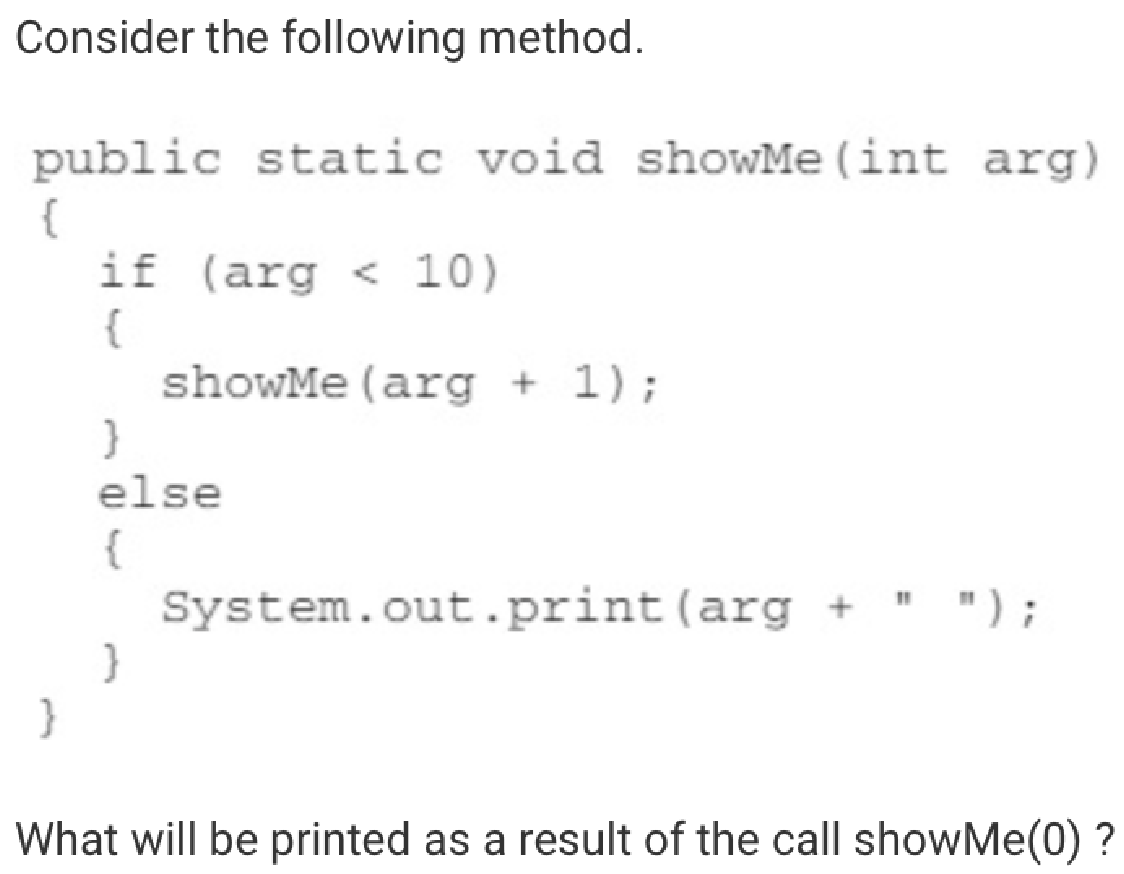
first when j=0 then result[0]=matrix[1][0]*matrix[0][2]; so result[0]=1*1=1 so result[0]=1

second when j=1 then result[1]=matrix[1][1]*matrix[1][2]; so result[1]=2*3=6 so result[1]=6

first when j=2 then result[2]=matrix[1][2]*matrix[2][2]; so result[2]=3*1=1 so result[2]=3

first when j=3 then result[3]=matrix[1][3]*matrix[3][2]; so result[3]=4*1=1 so result[3]=4

Question 15



Consider the following method.

```
public static void showMe(int arg)
{
  if (arg < 10)
  {
    showMe(arg + 1);
  }
  else
  {
    System.out.print(arg + " ");
  }
}
```

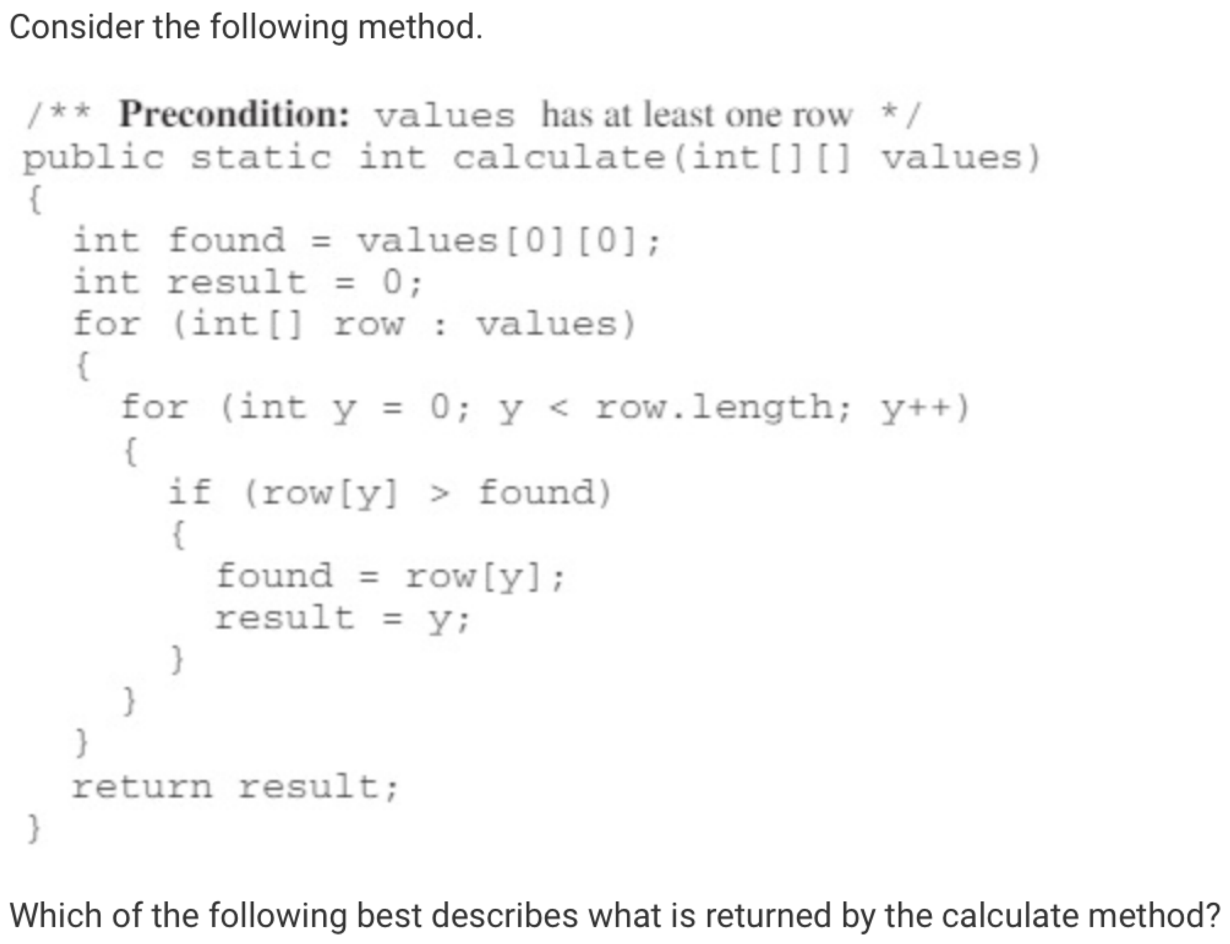
What will be printed as a result of the call showMe(0) ?

My Answer: 11
Correct Answer: 10

Explanation:

This is not a dumb mistake by me. This was just an array that moves up the next value. It’s not numerically adding 1.

Question 16



Consider the following method.

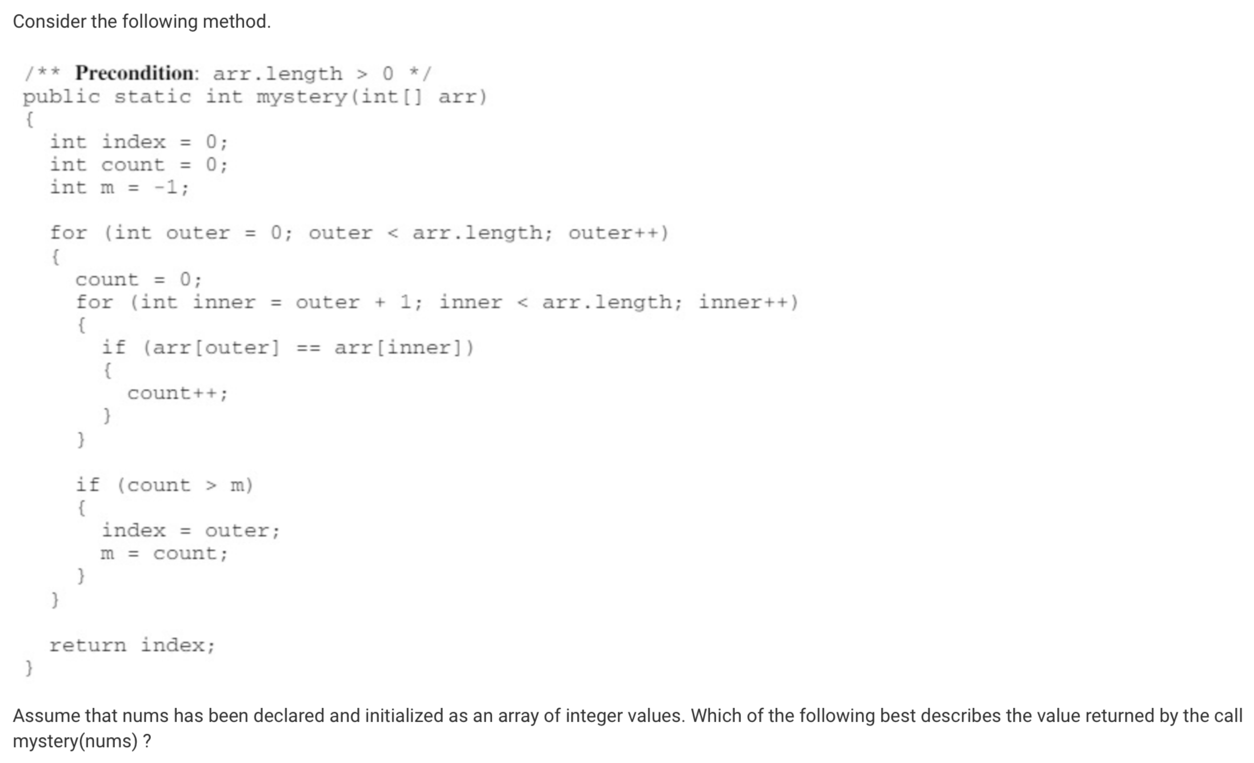
```
/** Precondition: values has at least one row */
public static int calculate(int[][] values)
{
  int found = values[0][0];
  int result = 0;
  for (int[] row : values)
  {
    for (int y = 0; y < row.length; y++)
    {
      if (row[y] > found)
      {
        found = row[y];
        result = y;
      }
    }
  }
  return result;
}
```

Which of the following best describes what is returned by the calculate method?

My Answer: The row index of an element with the smallest value in the two-dimensional array
Correct Answer: The column index of an element with the largest value in the two-dimensional array

Explanation: This was a wrong interpretation by me. It would return the largest value due to the if statement if (ro2[y] > found) continuously finds the next largest one. And then it’s columns because it searches by rows and compares the largest one per row which returns it to the column.

Question 20



Consider the following method.

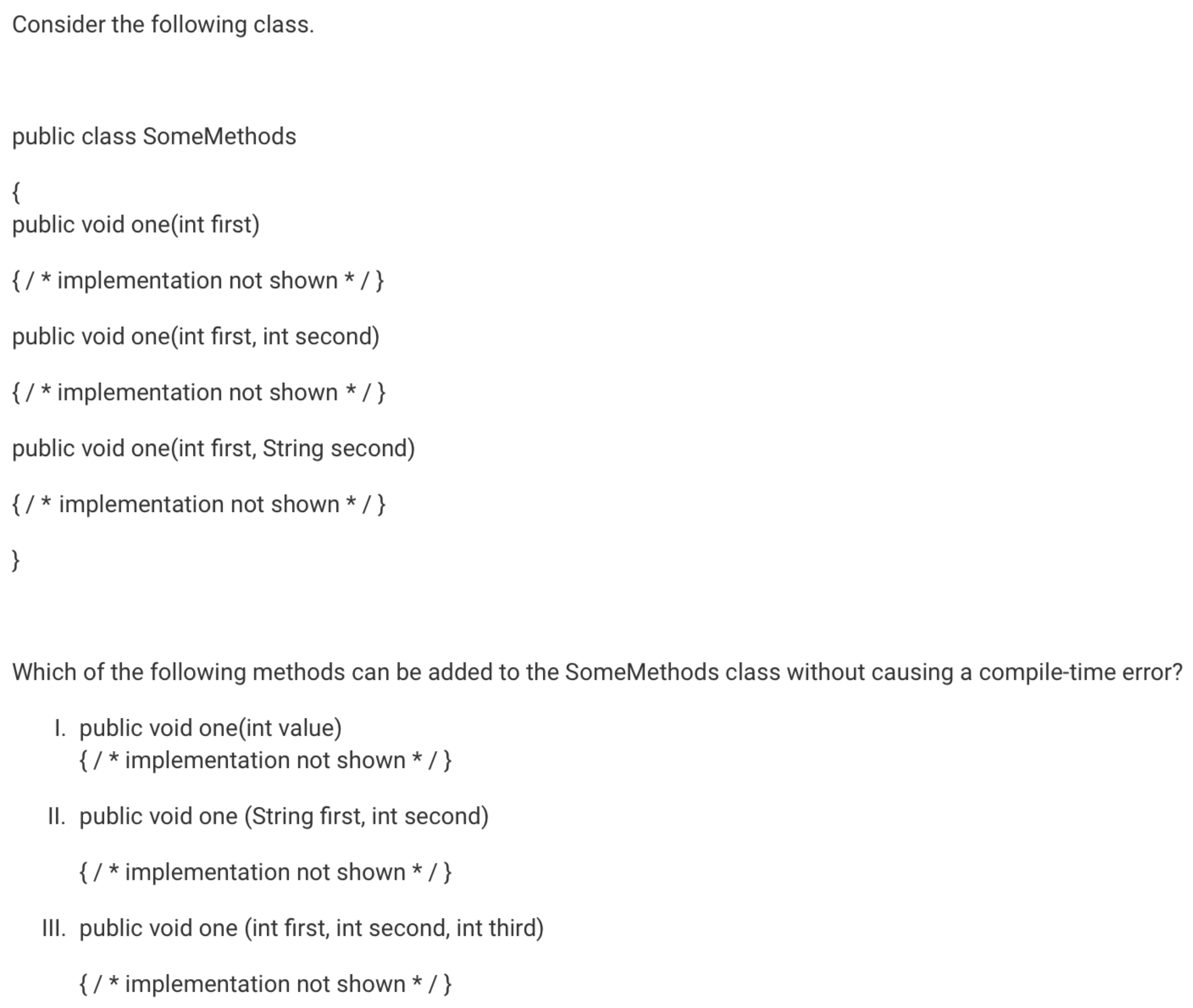
```
/** Precondition: arr.length > 0 */
public static int mystery(int[] arr)
{
  int index = 0;
  int count = 0;
  int m = -1;

  for (int outer = 0; outer < arr.length; outer++)
  {
    count = 0;
    for (int inner = outer + 1; inner < arr.length; inner++)
    {
      if (arr[outer] == arr[inner])
      {
        count++;
      }
    }

    if (count > m)
    {
      index = outer;
      m = count;
    }
  }

  return index;
}
```

Assume that nums has been declared and initialized as an array of integer values. Which of the following best describes the value returned by the call mystery(nums) ?

My Answer: A value that occurs most often in nums
Correct Answer: An index of a value that occurs most often in nums

Explanation: This is the index value because it returns index not the number.

Question 24



Consider the following class.

```
public class SomeMethods

{
public void one(int first)

{ / * implementation not shown * / }

public void one(int first, int second)

{ / * implementation not shown * / }

public void one(int first, String second)

{ / * implementation not shown * / }

}
```

Which of the following methods can be added to the SomeMethods class without causing a compile-time error?

I. `public void one(int value)`
   `{ / * implementation not shown * / }`

II. `public void one (String first, int second)`

   `{ / * implementation not shown * / }`

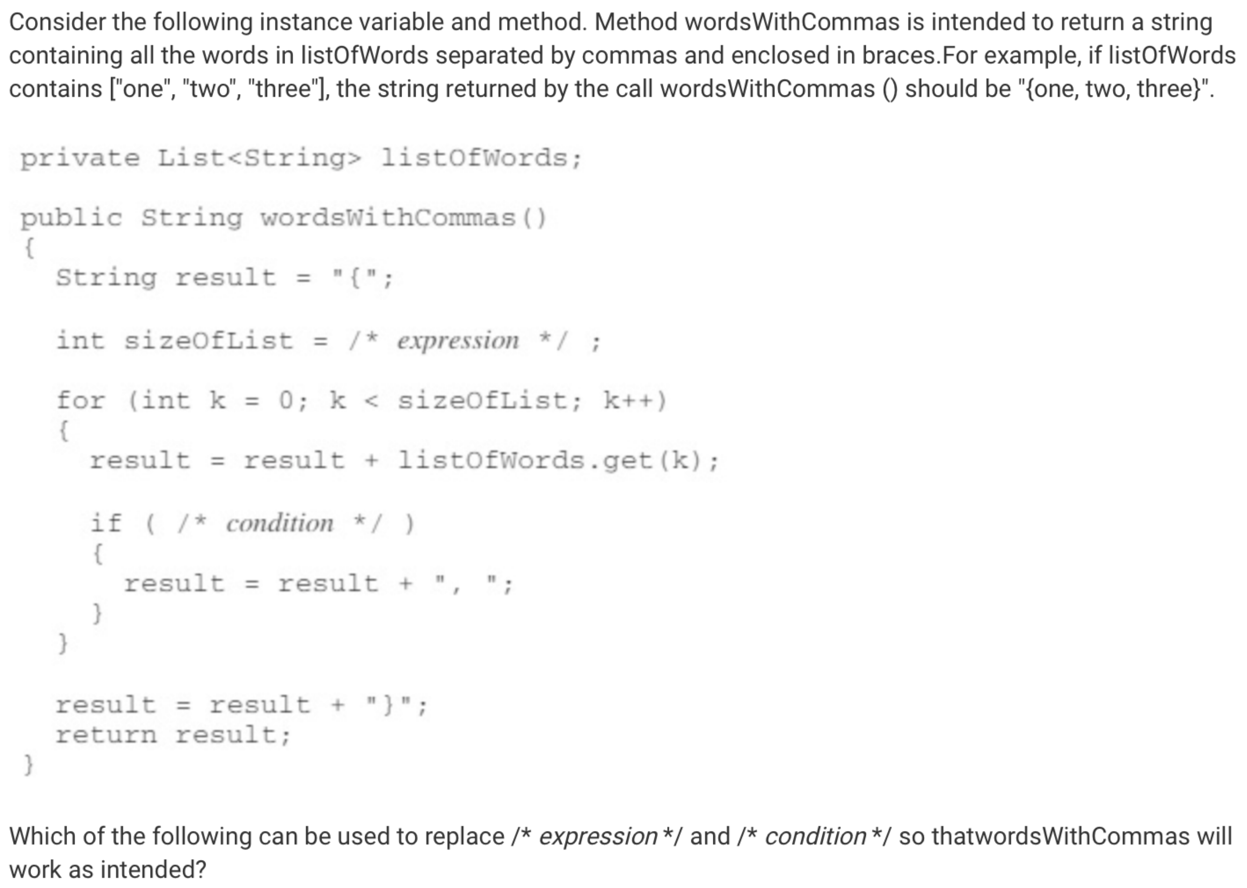
III. `public void one (int first, int second, int third)`

   `{ / * implementation not shown * / }`

My Answer: I, II, and III
Correct Answer: II and III only

Explanation: `public void one(int value)`
`{ / * implementation not shown * / }`
This does not work because there needs to be more parameters and if you don't put the second parameter it won't know which method to run first and which to run second, which would result in a compilation error.

Question 34



Consider the following instance variable and method. Method wordsWithCommas is intended to return a string containing all the words in listOfWords separated by commas and enclosed in braces.For example, if listOfWords contains ["one", "two", "three"], the string returned by the call wordsWithCommas () should be "{one, two, three}".

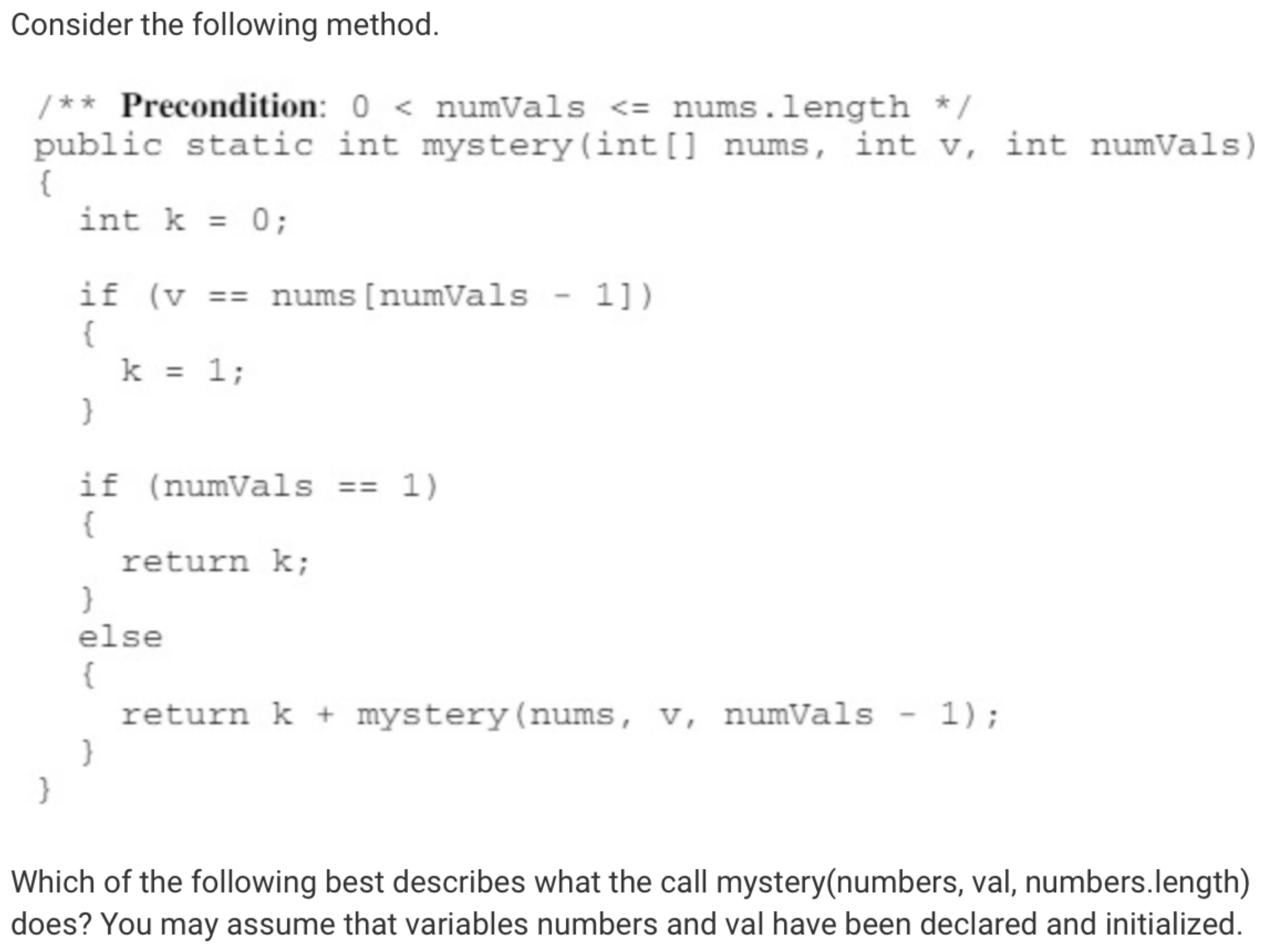
```
private List<String> listOfWords;

public String wordsWithCommas()
{
  String result = "{";

  int sizeOfList = /* expression */ ;

  for (int k = 0; k < sizeOfList; k++)
  {
    result = result + listOfWords.get(k);

    if ( /* condition */ )
    {
      result = result + ", ";
    }
  }

  result = result + "}";
  return result;
}
```

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

My Answer: / * *expression* * / **/** / * *condition* * /
listOfWords.size() - 1 / k != 0
Correct Answer: / * *expression* * / **/** / * *condition* * /
result.length() **/** k != 0

The latter is correct because if we established the list of word size before the line the code would result in a compelling error, whereas the result length parameter could be established afterwards and cause no compelling error.

Question 38



Consider the following method.

```
/** Precondition: 0 < numVals <= nums.length */
public static int mystery(int[] nums, int v, int numVals)
{
  int k = 0;

  if (v == nums[numVals - 1])
  {
    k = 1;
  }

  if (numVals == 1)
  {
    return k;
  }
  else
  {
    return k + mystery(nums, v, numVals - 1);
  }
}
```

Which of the following best describes what the call mystery(numbers, val, numbers.length) does? You may assume that variables numbers and val have been declared and initialized.

My answer: Returns the index of the last element in numbers that is equal to val
Correct answer: Returns the number of elements in numbers that are equal to val

Explanation: This question, unlike question 20, returns a specific number:
Return k + mystery(nums, v, numVals-1); returns a numeric value. Not the index.