Quiz 2 Unit 5 **AP CS** **Name** KEY scan up

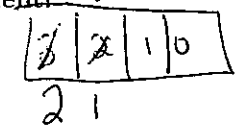
1. Consider the following method.

```
public void reduce(int[] arr, int len) {
        for (int k=0; k<len; k++)
        { arr[k]--; }
        len--;
```

len=3 local
K=0 1 2

What is the output of the following code segment?

```
int[] counts = {3, 2, 1, 0}
int len = 3;
reduce(counts, len);

for (int c : counts) {
        System.out.print(c+ " ");
}
System.out.println(len);
```



len: 3

↑ primitive never changes sending into method

A. 2 1 0 -1 2
B. 2 1 0 -1 3 ←
C. 2 1 0 0 2
D. 2 1 0 0 3 ←
E. 2 1 1 0 3 ←

these 3 only

Ms. goode's bad

2. A two-dimensional array *image* holds brightness values for pixels (picture elements) in an image. The brightness values range from 0 to 255. Consider the following method.

```
public int findMax(int[][] image) {
        int[] count = new int[256];          //line 2
        int i, iMax = 0;

        for (int r=0; r<image.length; r++) {
                for (int c=0; c<image[0].length; c++) {
                        i = image[r][c];
                        count[i]++;
                } //end inner for loop
        }
        for (i=0; i<256; i++) {
                if (count[i] > count[iMax])
                        iMax = i;
        } //end for loop
        return iMax;                         // line 15
} //end findMax
```

What does this method compute? actually returns index for

a) The column with the highest sum of brightness values in image.
b) The maximum brightness value for all pixels in image.
c) The most frequent brightness value in image.
d) The max sum of brightness values in an 256 by 256 square in image.
e) The max sum of brightness values in any 256 consecutive rows in image.

3. a) In question 2, please write what line 2 does?

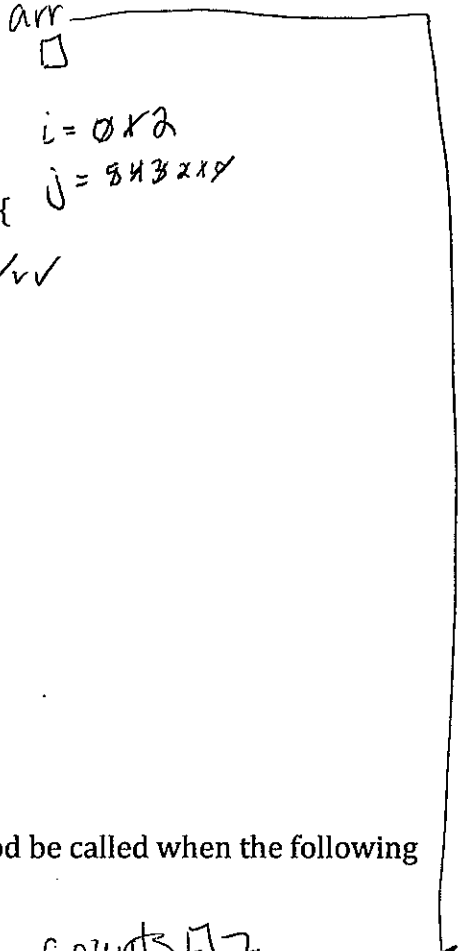
declares, initializes array of length 256 w/ all zeros.

b) In question 2, what is necessary to use line 15? (you may refer to a line in this code)

you must be in an accessor method of same type as variable returned. iMax should be guaranteed a value.

4. Consider the following class.

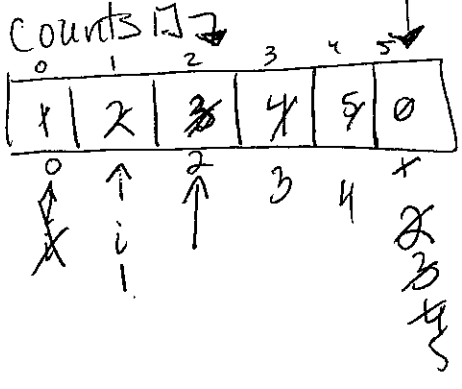
```
public class ArrayProcessor {

    public static void run(int[] arr) {

        //outer for loop begin
        for (int i=0; i<arr.length; i++) {

            for (int j=arr.length-1; j>i; j--) {
                if (arr[j] < arr[i])
                { swap(arr, i, j); }
            } //end inner for

        } //end outer for

    } //end method run

} //end class

private static void swap(int[] arr, int i, int j) {
    int temp = arr[i];
    arr[i] = arr[j];
    arr[j] = temp;
} //end swap
```



How many times will ArrayProcessor's swap method be called when the following code segment is executed? (be careful)

```
int[] counts = {1, 2, 3, 4, 5, 0};
ArrayProcessor.run(counts);
```

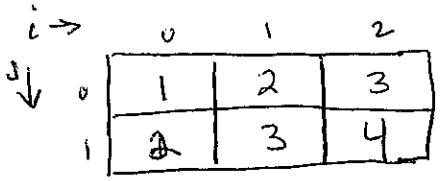
a) 1
b) 5
c) 15
d) 30
e) 35



5. Consider the following code segment.

rows cols

```
int[][] t = new int[2][3];
for (int i=0; i<t.length; i++) {
        for (int j=0; j<t[0].length; j++) {
                t[i][j] = i + j + 1;
        } //end inner for loop
} //end outer for loop
```



i = 0
j = 0

What is the result when the code segment is executed?

c) t holds the values 1 2 3
4 5 6

d) t holds the values 1 2 0
2 3 0

e) t holds the values 1 2 3
2 3 4

f) t holds the values 3 4 5
4 5 6

g) ArrayIndexOutOfBoundsException

6. Consider the following class and its constructor. You will finish the constructor according to the comments included.

```
public class Matrix {
        private int[][] m;

        /** Matrix constructor - Initializes m to a square n by n array with all the
            elements on the diagonal m[k][k] equal to 0 and all other elements equal
            to 1. **/

        public Matrix(int n) {
                m=new int[n][n];    // already all are zeros

                << add missing code here >>
                for (int i = 0; i < m.length; i++) {
                        for (int j=0; j< m[0].length; j++) {
                                if (i != j)
                                        m[i][j] = 1;
                        }
                }

        } //end constructor
} //end class Matrix
```

tough one

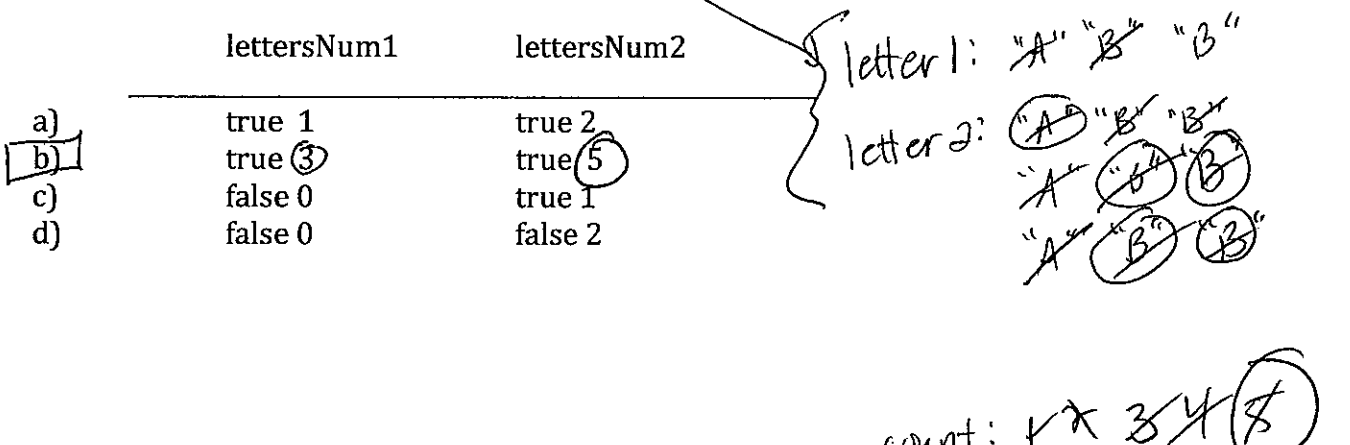
7. Consider the following method.

```
public boolean examine(String[] letters){
        int count = 0;
        for (String letter1 : letters){
                for (String letter2 : letters)
                {
                        if (letter1.equals(letter2))
                          count++;
                } // end inner for
        } //end outer for
        return count > 0;
} //end examine
```

What will examine return if examine is called with each of the following arrays as parameters? Include the count as well.

String[] lettersNum1 = {"A", "B", "C"};

String[] lettersNum22 = {"A", "B", "B"};



	lettersNum1	lettersNum2
a)	true 1	true 2
b)	true 3	true 5
c)	false 0	true 1
d)	false 0	false 2

note letter1 holds only "A then "B" then "C" while letter2 holds each letter 3 times! (bc inner loop is repeated for each of the letters in outer loop)