Problem Set 2, Part I

Problem 1: Variable scope

1) e, i
2) e, i, a, j, b
3) e, i, a
4) e, i, y
5) c
6) c, d

// "" + s1.charAt(6) + s1.substring(9) + " " + s2.substring(3, 8)
// Note that s2.substring(3) slices everything after its third position

Problem 2: String objects and their methods

2-1

a) `s1.substring(6) + " " + s2.substring(0, 2)`
b) `s1.substring(6, 7) + s1.substring(9) + " " + s2.substring(3)`
c) `s1.toUpperCase().substring(0, 1) + s1.toUpperCase().substring(9) + s2.substring(s2.length()-1)`
d) `s1.substring(0, 1) + s1.charAt(9) + s2.substring(0, 2)`
e) `s1.charAt(8)`
f) `s1.substring(8, 9)`
g) `s1.substring(0, 1) + s2.charAt(0)`
h) `s1.indexOf("i")`
i) `s1.substring(0, 4) + s1.replace('d', 'u').charAt(0) + " " + s1.charAt(6) + s1.replace('d', 'u').charAt(0) + s1.substring(8)`

Problem 3: Understanding code that uses an array

3-1)

i	values
-	{0, 1, 2, 3, 4, 5, 6, 7}
1	{0, 0, 2, 3, 4, 5, 6, 7}
3	{0, 0, 2, 2, 4, 5, 6, 7}
5	{0, 0, 2, 2, 4, 4, 6, 7}
7	{0, 0, 2, 2, 4, 4, 6, 6}

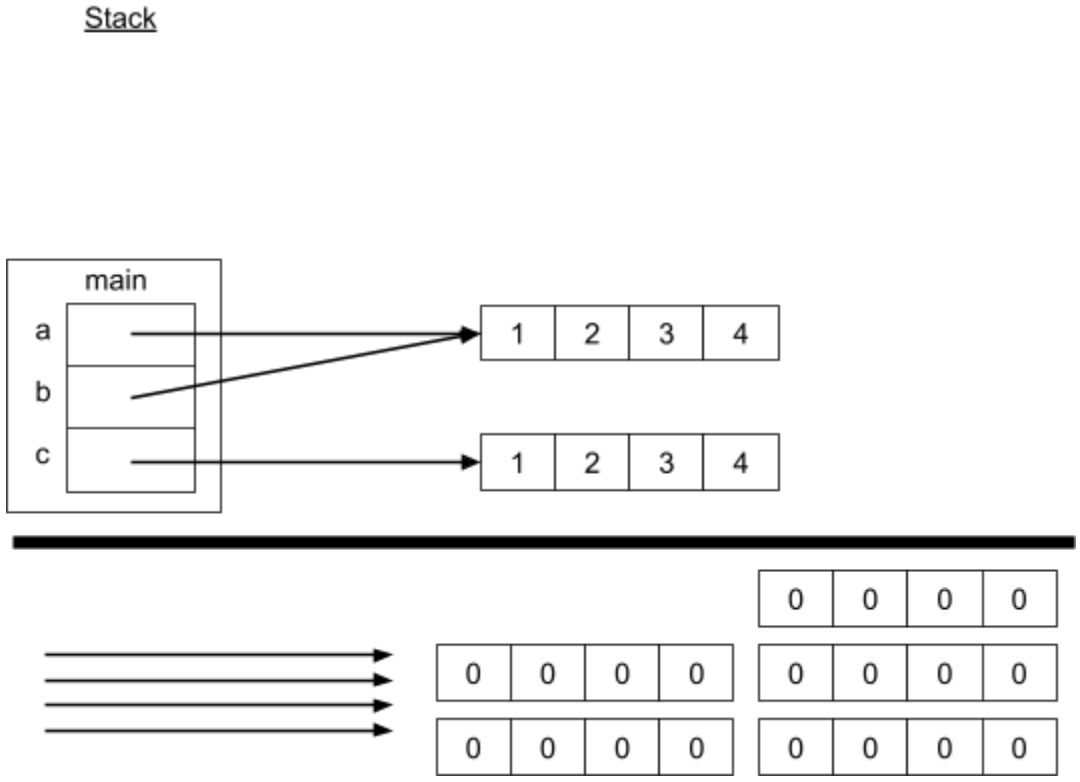
3-2) contents of array just before the method returns:

`{6, 0, 2, 2, 4, 4, 6, 6}`

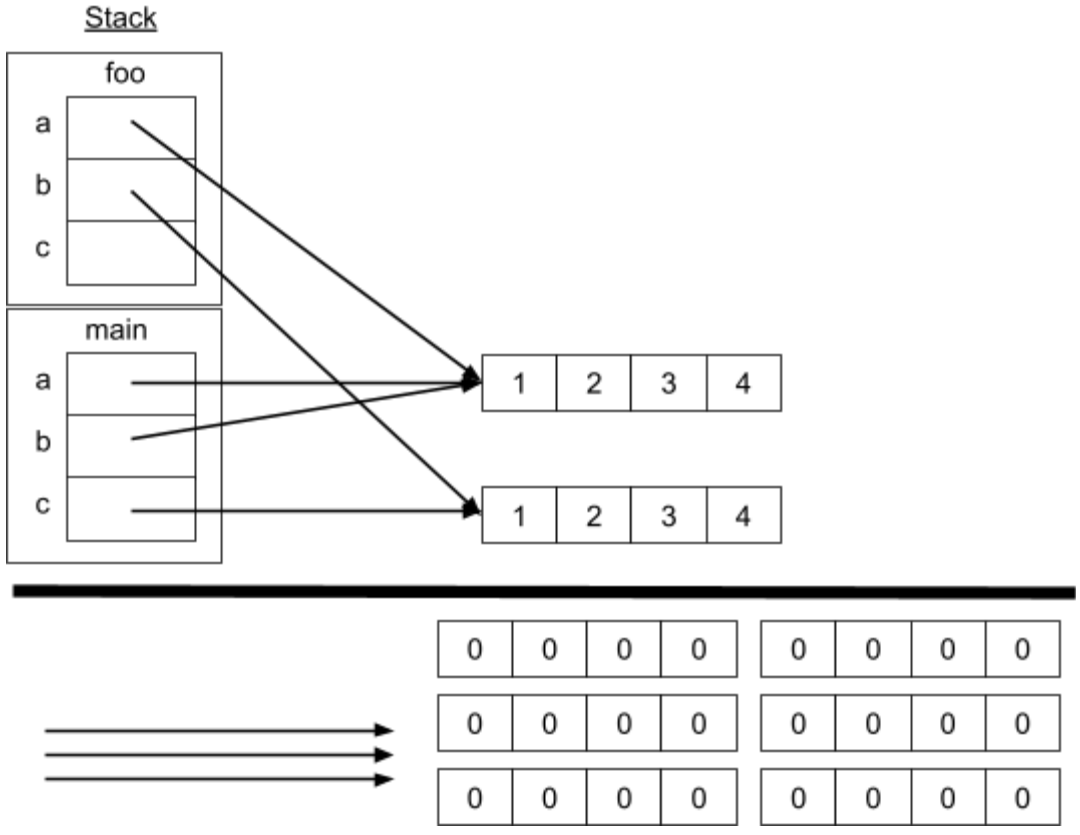
3-3) After this print statement, we will see the changes made by the call to the mystery() method, since those changes (i.e. item assignments) in the mystery() method are done to the same array and alter the "internals" of the array arr.

Problem 4: Arrays and memory diagrams

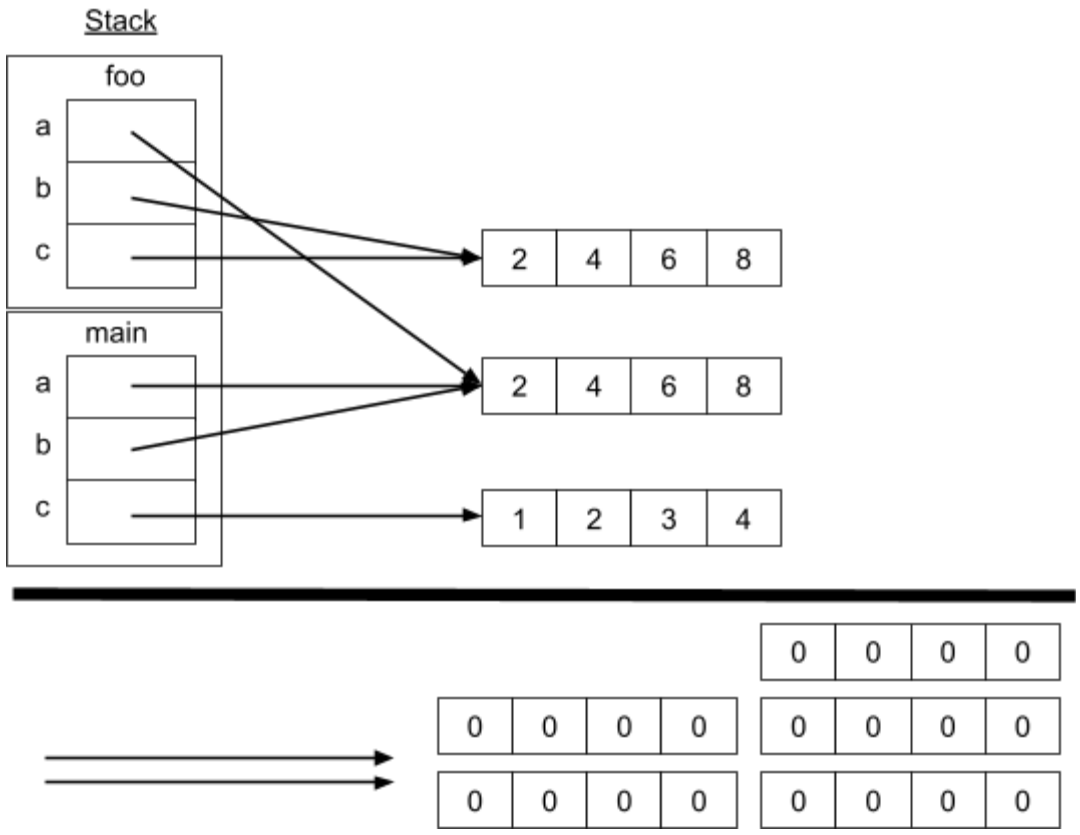
4-1)



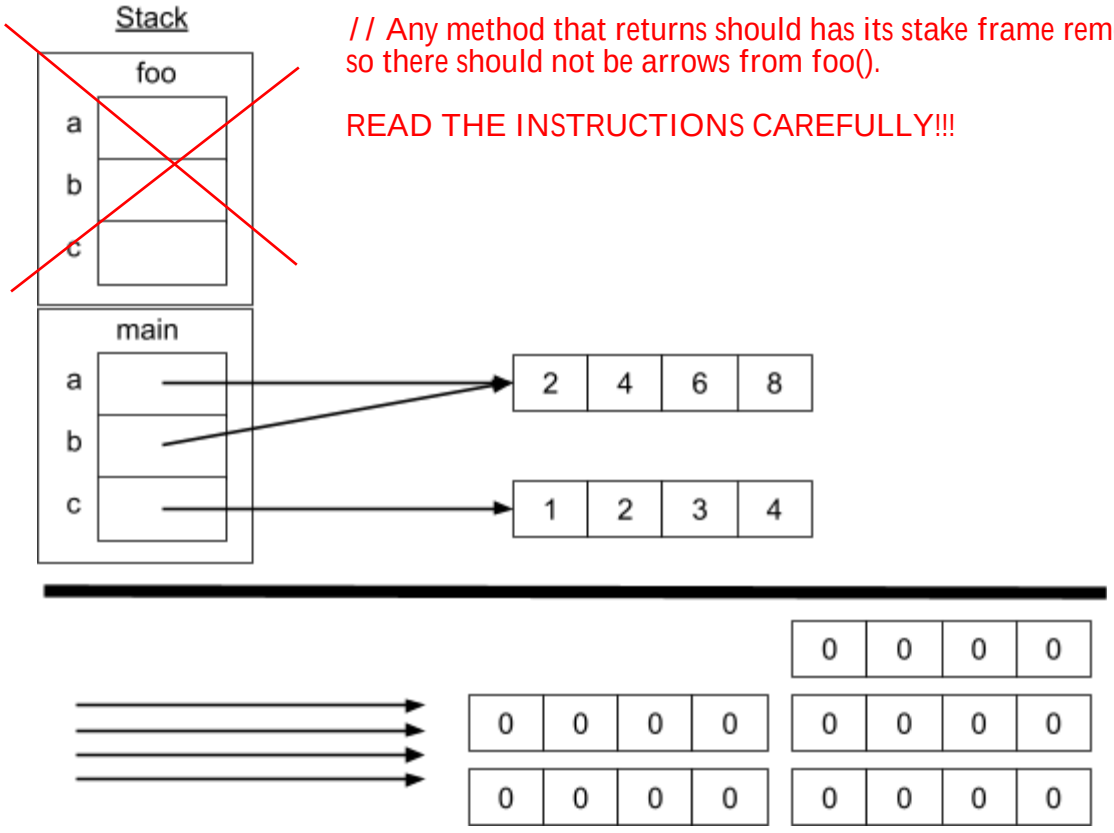
4-2)



4-3)



4-4)



// Any method that returns should has its stake frame removed, so there should not be arrows from foo().

READ THE INSTRUCTIONS CAREFULLY!!!

Problem 5: Two-dimensional arrays

5-1) twoD[2][1] = 30;

5-2)

```
for (int i = 0; i < twoD.length; i++) {
      for (int j = 0; j < twoD[0].length; j++) {
            if (j == twoD[0].length - 1) {
                  System.out.println(twoD[i][j]);
            }
      }
}
```

5-3)

```
for (int i = 0; i < twoD.length; i++) {
      for (int j = 0; j < twoD[0].length; j++) {
            if (i == j) {
                  System.out.println(twoD[i][j]);
            }
      }
}
```