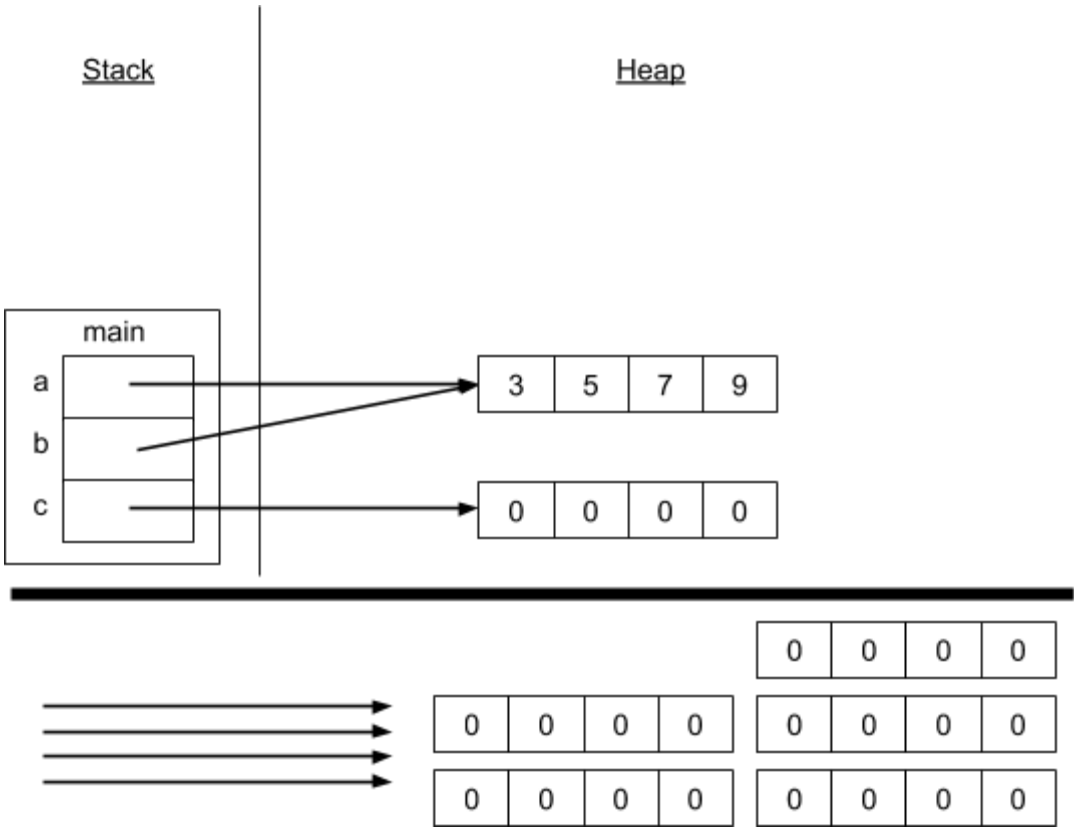
Problem Set 2, Part I

Problem 1: Understanding code that uses an array
1-1)

i	val1	val2	arr
-	-	-	{1, 3, 5, 7, 9, 11, 13}
0	3	5	{8, 3, 5, 7, 9, 11, 13}
1	5	7	{8, 12, 5, 7, 9, 11, 13}
2	7	9	{8, 12, 16, 7, 9, 11, 13}
3	9	11	{8, 12, 16, 20, 9, 11, 13}
4	11	13	{8, 12, 16, 20, 24, 11, 13}

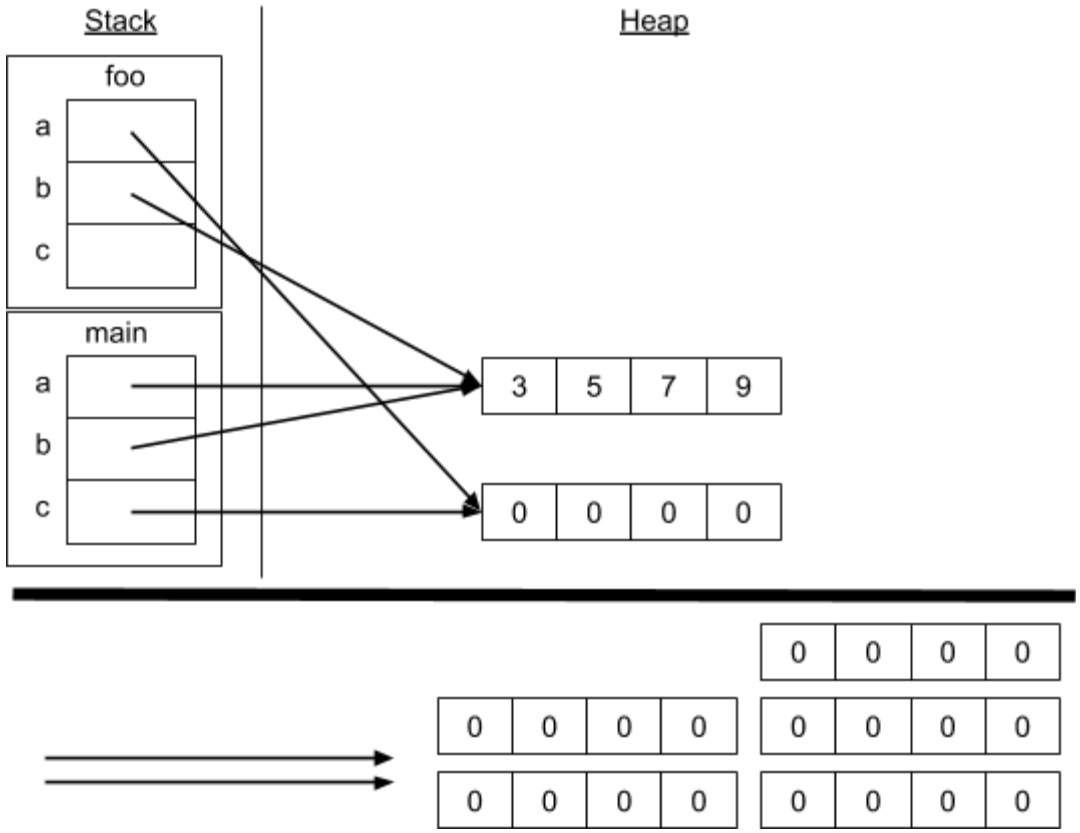
1-2) I see the array that changed by the call to the mystery(). When you make the array, the array variable does not store the array itself. It stores a reference to the array(the memory address of the array). So when u change the value of the array, you are changing the reference address, which is stored.

Problem 2: Memory management and arrays
2-1)

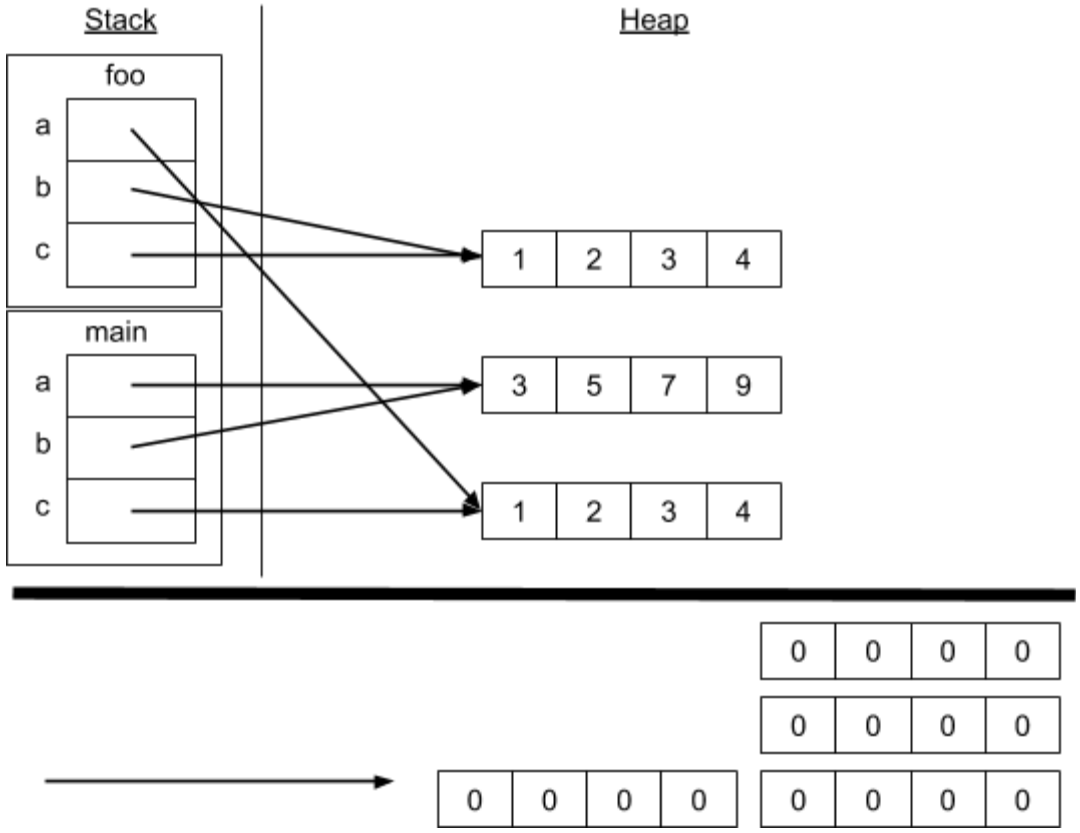

Stack
Heap
main
a
b
c
3
5
7
9
0
0
0
0

0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0

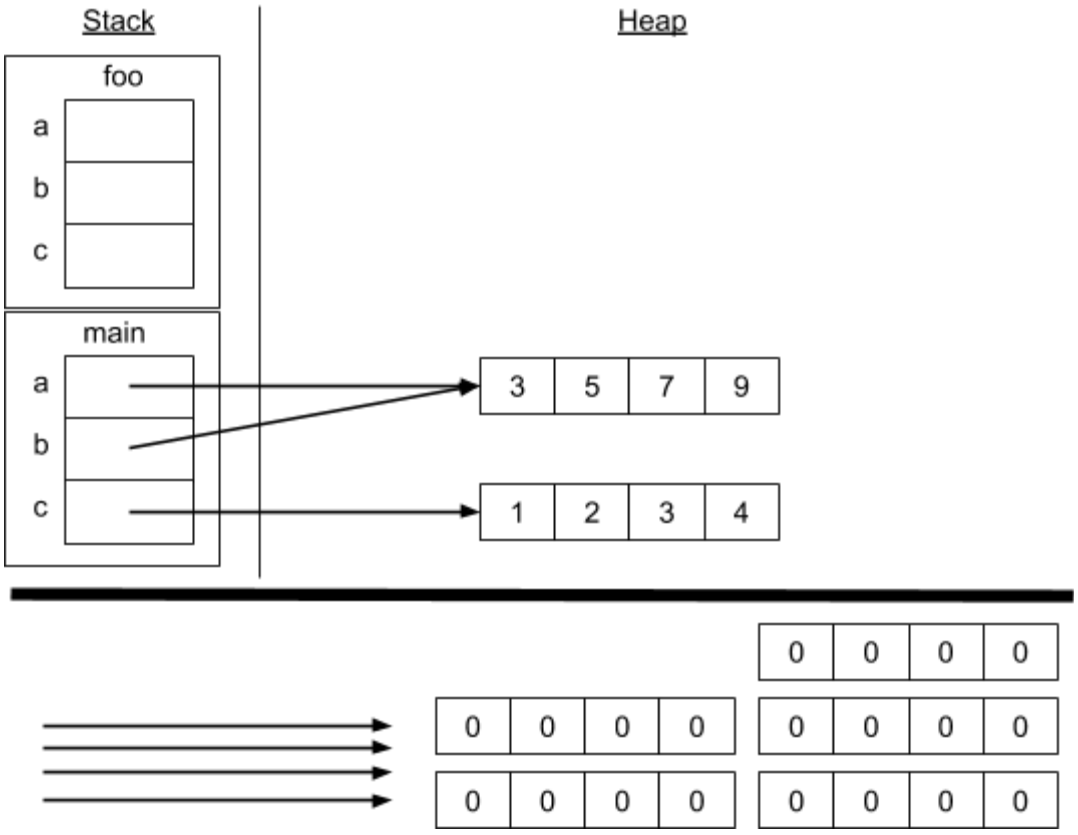
2-2)


Stack
Heap
foo
a
b
c
main
a
b
c
3 5 7 9
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0

2-3)


Stack
Heap
foo
a
b
c
main
a
b
c
1 2 3 4
3 5 7 9
1 2 3 4
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0

2-4)



Problem 3: Two-dimensional arrays

3-1) twoD[2][0] *= 2;

3-2)
```
int col = twoD[0].length;
    for(int i = 0; i < twoD.length; i++){
        System.out.println(twoD[i][col - 1]);
    }
```

3-3)
```
for(int i = 0; i < twoD.length; i++){
        System.out.println(twoD[i][i]);
    }
```

Problem 4: Our Rectangle class revisited

4-1)

type of method: mutator

Header: **public Void** rotate()

4-2)

type of method: accessor

Header: **public boolean** largerThan(Rectangle other)

4-3)

problems in code: width and height is encapsulation by putting “private” in front of the code in fields, but r1.width = r1.width + 20; → tried to change the set value

rewritten version: r1.grow(20, 0) → Mutator method