Problem Set 2, Part I

Problem 1: Variable scope

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

Problem 2: String objects and their methods

2-1

```
a) str2.substring(0, 5) + str1.substring(5)
b)”b”+str2.substring(8)
c)str1.substring(0,5).toLowerCase + str2.substring(6).toUpperCase()
d)str1.charAt(4)
e)str2.charAt(2) + str2.charAt(11)
f)str1.indexOf(‘r’) + str1.indexOf(‘t’)
g)str1.substring(0,2).toLowerCase()+str2.charAt(2) +
str1.substring(3,11) + str2.charAt(2) + str1.substring(12)
```

Problem 3: Understanding code that uses an array

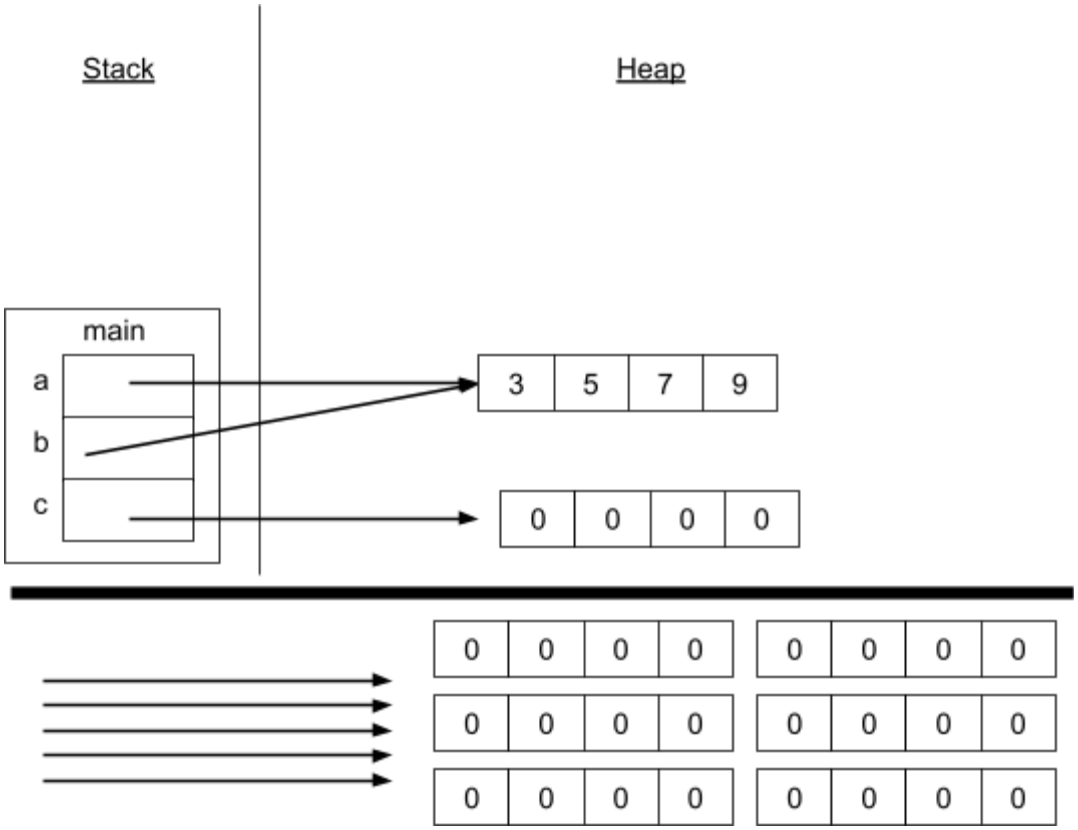
3-1)

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}

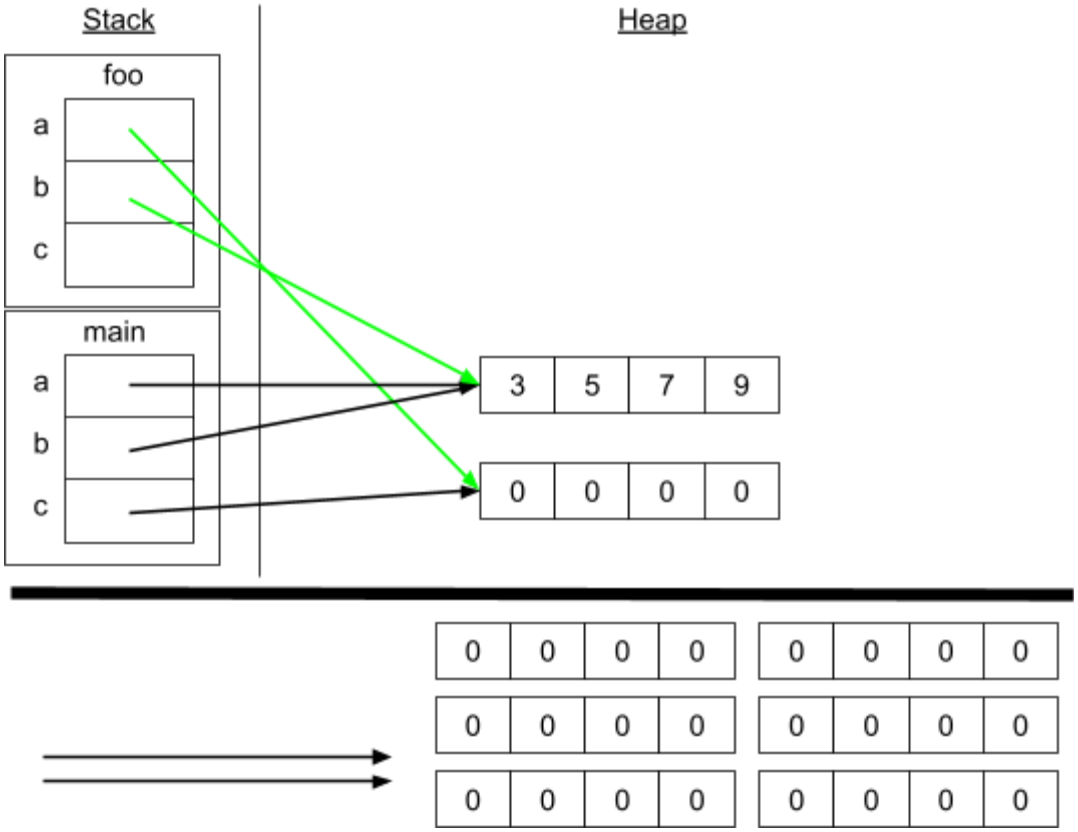
3-2) We do not see the original array, because the for loop was going into the index and changing the value at that location. Since the value isn’t stored in the stack, the index of each array was referencing the location in the heap, which is what we changed, the value. This is why when you print the array only the location will be printed, and not the value–unless you called the Arrays class.

Problem 4: Memory management and arrays

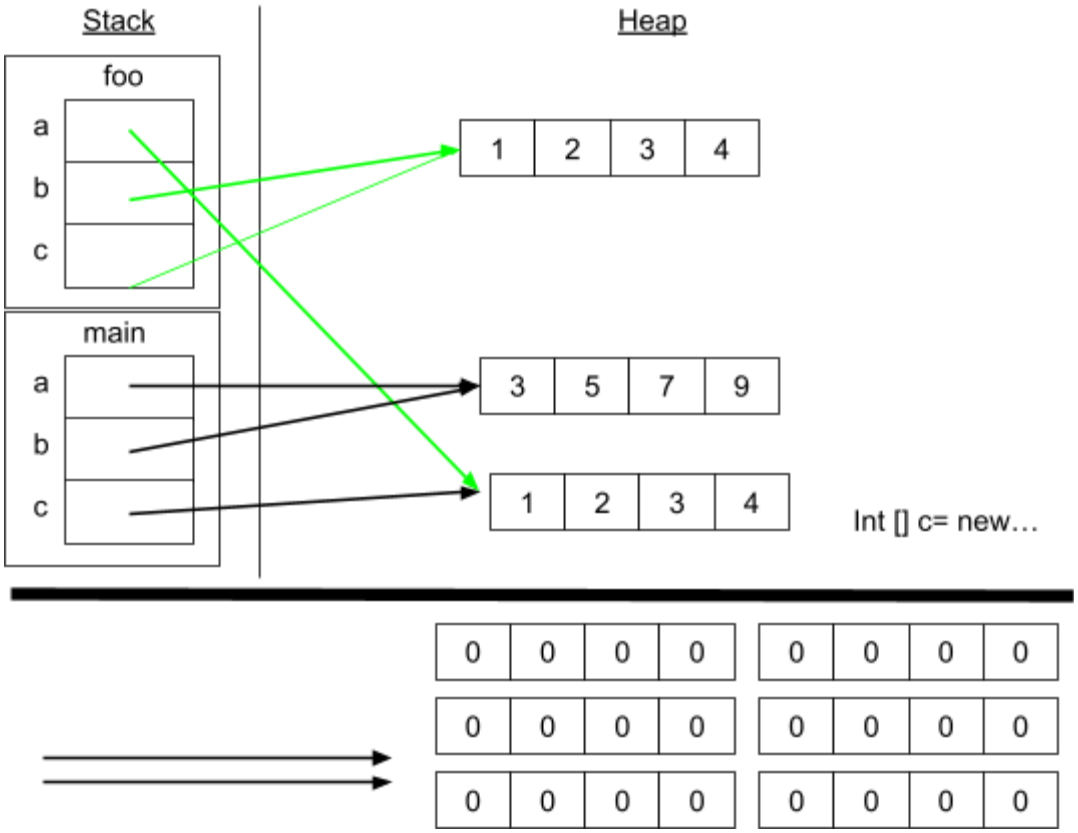
4-1)



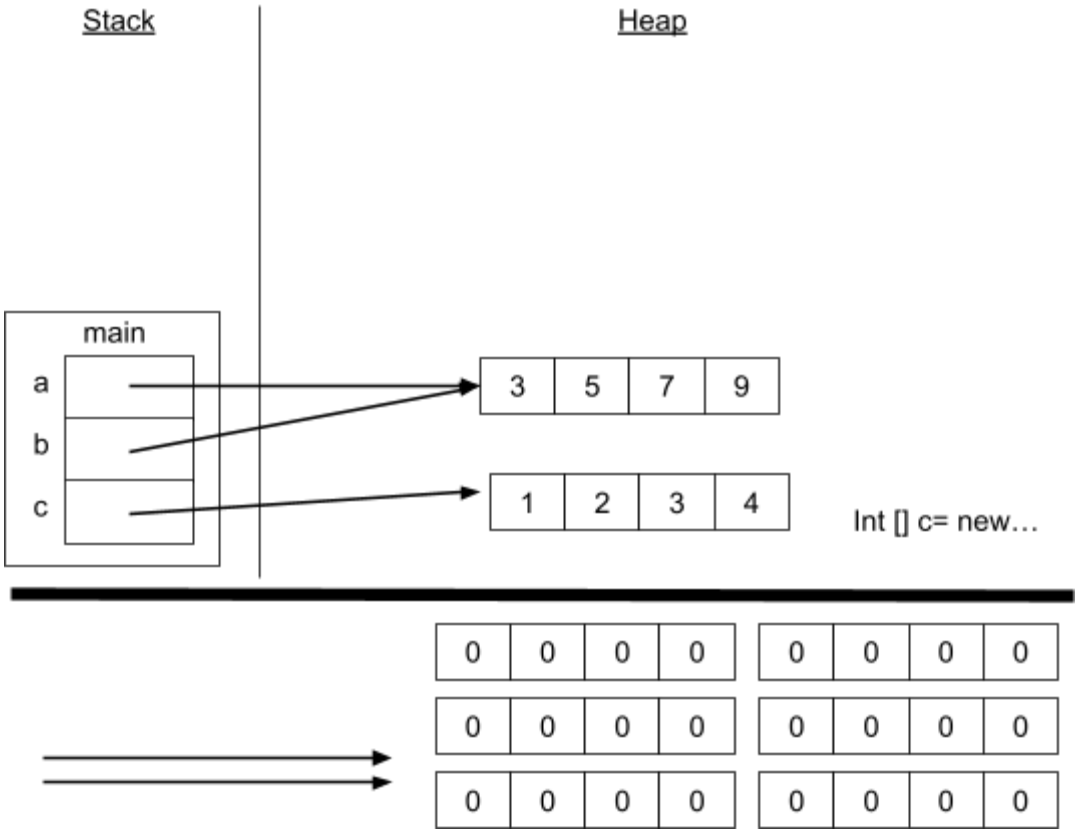
4-2)



4-3)


Stack
Heap
foo
a
b
c
main
a
b
c
1 2 3 4
3 5 7 9
1 2 3 4
Int [] c= new…
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0

4-4)



Problem 5: Two-dimensional arrays

5-1) twoD[1][2] = 14;

5-2) for(int i = 0; i < twoD.length;i++)
System.out.println(twoD[i][0]);

5-3) for(int i = twoD.length-1; i > -1; i- -){
for(int j = twoD.length-1; j > -1; j- -){ //its a square array so the # col = # rows
if (i+j == 3){ // antidiagonals have the same value row+column values
System.out.println(twoD[i][j]);
} else continue;
`}
}