Problem 3:

In this written response, I'll number the sections of my explanation which will correspond to the numbers in the accompanying picture, located on the last page (not every section will have a corresponding number). Also, in my written response and in my picture, I'll refer to variables in functions with the addition of "Func" added to the end of their original names.

(1): First, 3 ints named i, j, and k are declared and assigned the values 8, 1, and 0, respectively.

(2): Next, the function f is called and given the arguments j and k. This introduces a function scope and initializes the parameters iFunc and jFunc. iFunc is declared as a reference to the int j, the first argument, and jFunc is declared as an int and assigned 0, which is the value that the second argument k contains.

(3): Next, an int named tmp is declared and assigned the value iFunc. iFunc is a reference to j, which contains the value 1, so tmp is assigned the value 1.

(4): Next, iFunc is assigned the value jFunc. jFunc contains the value 0, and iFunc is a reference to j, so this means j is assigned the value 0.

Next, an if statement asks whether tmp equals 8. The result is false, so the code inside the braces does not run.

(5): Next, jFunc is assigned the value tmp. tmp contains the value 1, so jFunc is assigned the value 1.

(6): Next, "return jFunc" is encountered. jFunc contains the value 1, so 1 is returned (this doesn't do anything, as the function call wasn't assigned to anything). In addition, the function call ends and the function scope is destroyed.

(7): Next, i, a space, j, a space, k, and a newline character are printed to the console. i, j, and k currently hold the values 8, 0, and 0 respectively, so "8 0 0" is displayed on the console.

(8): Next, the function f is called and given the arguments i and j. This introduces a function scope and initializes the parameters iFunc and jFunc. iFunc is declared as a reference to the int i, the first argument, and jFunc is declared as an int and assigned 0, which is the value that the second argument j contains.

(9): Next, an int named tmp is declared and assigned the value iFunc. iFunc is a reference to i, which contains the value 8, so tmp is assigned the value 8.

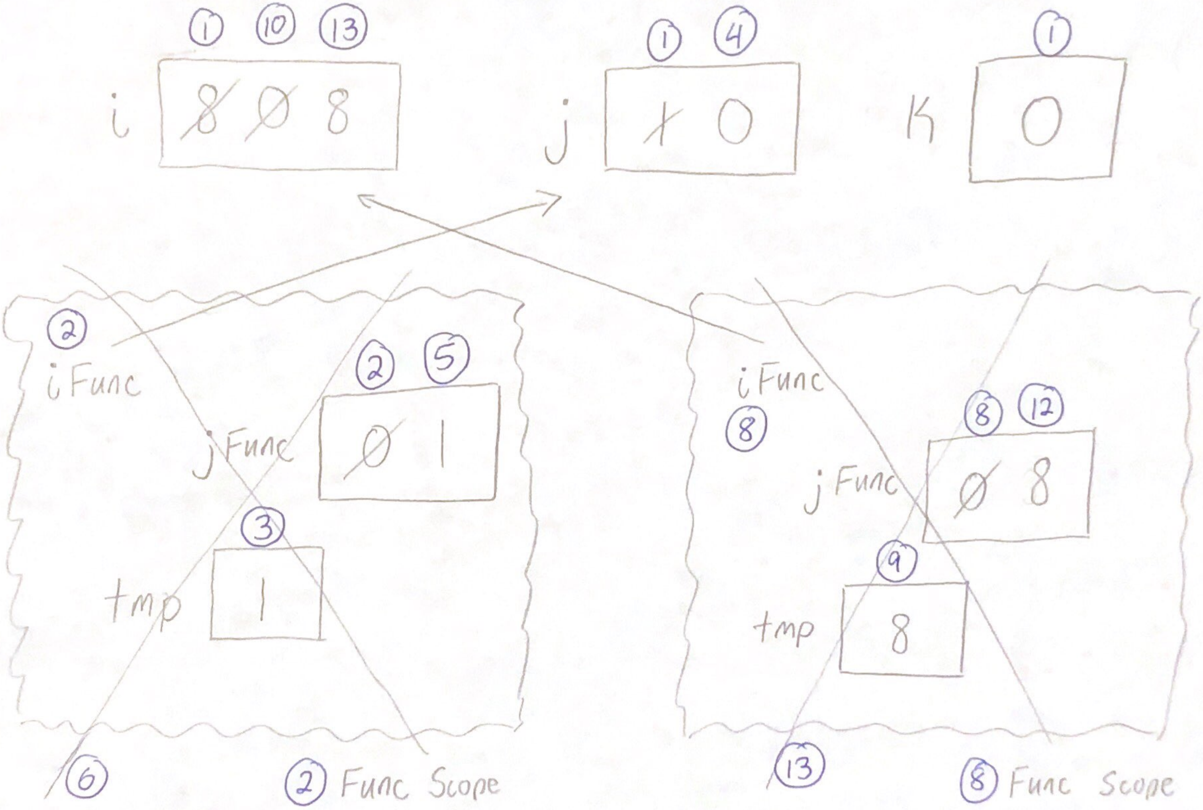
(10): Next, iFunc is assigned the value jFunc. jFunc contains the value 0, and iFunc is a reference to i, so this means i is assigned the value 0.

(11): Next, an if statement asks whether tmp equals 8. The result is true, so the code inside the braces runs: tmp, a space, iFunc, another space, jFunc, and a new line character are printed to the console. tmp, and jFunc currently hold the values 8 and 0 respectively, and iFunc is a reference to i, which currently holds the value 0, so “8 0 0” is displayed on the console.

(12): Next, jFunc is assigned the value tmp. tmp contains the value 8, so jFunc is assigned the value 8.

(13): Next, “return jFunc” is encountered. jFunc contains the value 8, so 8 is returned. i was assigned the function call, so this means i is assigned the value 8. In addition, the function call ends and the function scope is destroyed.

(14): Finally, i, a space, j, a space, k, and a newline character are printed to the console. i, j, and k hold the values 8, 0, and 0 respectively, so “8 0 0” is displayed on the console.



⑦ "8 0 0" is printed to the console.

⑪ "8 0 0" is printed to the console.

⑭ "8 0 0" is printed to the console.