PART I

a) The execution will continue after the call to the function.

b) A name of a function has the same restrictions upon it as naming a variable. Must begin with a letter, is case sensitive, can contain digits, letters, underscores, and dollar signs.

c)

1. function onePlusOne() {

//your code in here

}

2. function twoPlusTwo() {

//your code in here

}

d) How accessible is it. Is it only accessible inside this function? Is it accessible everywhere?

e) A local variable is a variable that can only be accessed within the function that is currently being run/executed.

f) void

g) This is a function which it takes an input, and returns a value. This function will do nothing else but this, it will not adjust the value of the input, or any other variables. It simply does one task.

h) The purpose of the call stack is so that the system is able to perform the recursive function at all. The system will use the stack as a tracker for what to execute next when running a recursive function. FILO policy.

i) It allows a much cleaner and easier way to organize code, and to compartmentalize/modulate the code by making it more modular. This way there are small tasks being handled by more functions.

j) A recursive call is a call when a function calls itself.

k) A function's parameter is/are the value(s) it takes in

l) Stack overflow happens when a program/function tries to use more memory than there is on the stack.

m) The code will not work as the function is expecting less variables than that. It will not know what to do with the extra one.

n) When a function too few arguments, it will also not work because it is waiting on one, or more variables, that are undefined.

o) One value.

p) It can have as many as the user needs by using if statements.

PART II

a) The function name is doubleOrTriple.

There is one argument (x), and (x) should be a number.

The possible return values are NaN, x\*2, x\*3.

This call will return 80

b) i = 6 j = 4 k = 6/4

i = 6 j = 8 k = 7

c) This code draws a shape with the center of the triangle defined in the function parameter.

ctx is the context of the canvas you want to draw on. x and y are the coordinates of where the center of your triangle is. N is the number of pointy edges that your star will have. r1 and r2 are how far from the centre to start drawing, and the lengths of the sides of the triangles.

The benefit of using parameters is that, now my function can be used anywhere on this canvas with any sort of edge lengths. All I will have to do is change what I when I call the function, and this could potentially be user-defined by using prompt.

drawSomething(ctx,250,250,100,100,50);

PART III

a) function returnMid(x, y, z) {

if((x > y && x < z) || (x < y && x > z)) {

return x;

} else if((x < y && y < z) || (x < y && y > z)) {

return y;

} else if((z > y && z < x) || (z < y) && (z > x)) {

return z;

}

}

b)

function findHandShakes(N) {

if(N == 1) {

return 0;

} else if(N == 2) {

return 1;

} else {

return findHandShakes(N-1)+N-1;

}

}