Homework #JSVIII

1. Explanations of terms as explained to a 12-year-old –

**Recursion** is when a function calls itself, over and over again in a way that will increment itself to reach a stopping point, and then it will jump back out of each level to the top level and then the function will be complete. It will resolve only after it has looped through this process and reached the stopping point and then looped back out to the level of the original function call. Here is a simple example of recursion, to add the sum of all the numbers from 1 to whatever positive value you put in to start.

const sum = function(x) {

let result = 0;

recursiveSum(x);

function recursiveSum(n) {

if (n < 1) {

return;

}

result = result + n;

recursiveSum(n – 1);

}

return result;

}

sum(5);

Here, we call the function with x = 5, and after setting our result to start at 0, then the recursive function is called with the value that we passed in for x. As long as the value that is passed into the function is greater than or equal to 1, then it will add the value to the result and then call itself again with the value decreased by one. It will keep looping through this and adding the values until the value is less than 1. That is the base case defined at the beginning of the recursive function to tell it when to jump out of the function and return. It returns back through each level until it reaches the top level, and then jumps out of the function returning the result. The result in this case is 15, which is 5 + 4 + 3 + 2 + 1.