**Recursion Worksheet**

For each of the following use the trace method taught to you in class to help identify what is the end result for each recursive function call:

1. For the following method, what would be displayed by the call *mystery1(5)*?

function mystery1(nNum)

{

if (nNum <= 0)

{

return 0;

}

else

{

return nNum + mystery1(nNum - 1);

}

}

15

2. For the following method, what would be displayed by the call *mystery2(5)*?

function mystery2(nNum){

if(nNum <= 0)

{

return 0+"";

}

return (nNum) + " " + mystery2(nNum - 1);

}

5 4 3 2 1 0

3. For the following method, what would be displayed by the call: *mystery3(4)*?

function mystery3(nNum){

let data="";

if(nNum <= 0)

{

return "";

}

for(let nI = 0; nI < nNum; nI++)

{

data+=("-");

}

for(let nI = 0; nI < nNum; nI++)

{

data+=("+");

}

return data + mystery3(nNum - 1) +"\n";

}

----++++---+++

--++

-+

4. For the following method, what value would be returned by the call: *ans =mystery4(4)?*

function mystery4(nNum)

{

if (nNum > 1)

{

return nNum \* mystery4(nNum - 2);

}

else

{

return 2;

}

}

16

5. For the following method, what value would be returned by the call *ans=mystery5(6,8)*?

function mystery5(k, n)

{

if (n == k)

{

return k;

}

else

{

if (n > k)

{

return mystery5(k, n - k);

}

else

{

return mystery5(k - n, n);

}

}

}**}**

2

6. For the following method, what would be displayed by the call: *mystery6(“abcdefgh”)*?

function mystery6(sWord){

let nL = sWord.Length;

if (nL > 1)

{

String sTemp = sWord.substring(Math.round(nL / 2.0);

return sTemp + mystery6(sTemp);

}

else { return ""; }

}

efghghh