***Recursion Worksheet Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

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

public void mystery1(int n){

if(n <= 0)

return; **5**

else **4**

{ **3**

System.out.println(n); **2**

mystery1(n - 1); **1**

}

}

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

public void mystery3(int n){

if(n <= 0)

return;

for(int i = 0; i < n; i++)

System.out.print("-");

for(int i = 0; i < n; i++)

System.out.print("+");

System.out.println();//ends the line

mystery3(n - 1);

}

----++++

---+++

--++

-+

D. For the following method, what would be displayed by the call: mystery4(4)?

public void mystery4(int n){

if(n <= 0)

return;

mystery4(n - 1);

for(int i = 0; i < n; i++)

System.out.print("-");

for(int i = 0; i < n; i++)

System.out.print("+");

System.out.println();//ends the line

}

-+

--++

---+++

----++++

E. For the following method, what would be displayed by the call: mystery5(“abcd”)?

public void mystery5(String sWord){

int nL = sWord.length();

if(nL > 1)

{

String sTemp = sWord.substring(0,nL-1);

System.out.println(sTemp);

mystery5(sTemp); **abc**

} **ab**

} **a**

F. For the following method, what would be displayed by the call: mystery7(“abcdefghijkl”)?

public void mystery7(String sWord){

int nL = sWord.length();

if(nL >= 3)

{

mystery7(sWord.substring(0,nL/3));

System.out.println(sWord.substring(0, nL/3));

mystery7(sWord.substring(0,nL/3));

//substring(x) same as substring(x,length())

}

}

a

abcd

a

G. For the following method, what would be displayed by the call: mystery8(“la-la-la”)?

public void mystery8(String sWord){

int nL = sWord.length();

if(nL >= 3)

{

mystery8(sWord.substring(0,nL/3));

System.out.println(sWord.substring(nL/3,2\*nL/3));

mystery8(sWord.substring(2\*nL/3));

//substring(x) same as substring(x,length())

}

}

-la

l

### Recursive Tracing

1. Consider the following method:

public int mystery(int x, int y) {

if (x % 2 == 1 || y % 2 == 1) {

return 1;

} else {

return 2 \* mystery(x / 2, y / 2);

}

}

For each call below, indicate what value is returned:

Method Call Value Returned

mystery(4, 19) 1

mystery(32, 56) 8

mystery(12, 20) 4

2. Consider the following method:

public int mystery1(int x, int y) {

if (x < y)

return x;

else

return mystery1(x - y, y);

}

For each call below, indicate what value is returned:

Method Call Value Returned

mystery1(6, 13) 6

mystery1(14, 10) 4

mystery1(37, 10) 7

3. Consider the following method:

public void mystery2(int n) {

if (n <= 1)

System.out.print(n);

else {

mystery2(n / 2);

System.out.print(", " + n);

}

}

For each call below, indicate what output is produced by the method:

Method Call Output Produced

mystery2(1) 1

mystery2(2) 1, 2

mystery2(3) 1, 3

mystery2(30) 1, 3, 7, 15, 30

mystery2(100) 1, 3, 6, 12, 25, 50, 100