edhesive

Worksheet: Tracing Recursion – Solutions

1. Consider the following recursive method:

```
public static void recur1(int n) {
      if (n > 0)
            recur1(n - 2);
      System.out.print(n + " ");
}
```

Trace the output for the following method calls:

a. `recur1(6);` 0246
b. `recur1(3);` -113
c. `recur1(5);` -1135
d. `recur1(0)` 0

2. Consider the following recursive method:

```
public static void recur2(int n) {
      if (n > 0) {
            recur2(n - 1);
            for (int i = 0; i < n; i++)
                  System.out.print("*");
            System.out.println();
      }
}
```

Trace the output for the following method calls:

a. `recur2(6)`

```
*
**
***
****
*****
******
```

b. `recur2(3);`

```
*
**
***
```

c. `recur2(1);`

```
*
```

3. Consider the following recursive method:

```
public static void recur3(int n) {
      if (n > 0)
            recur3(n - 10);
      System.out.print(n + " ");
}
```

Trace the output for the following method calls:

a. `recur3(50);` 0 10 20 30 40 50
b. `recur3(18);` -2 8 18
c. `recur3(-50);` -50

edhesive

4. Consider the following recursive method:

```
public static void recur4 (int n){
      if (n > 1)
            recur4(n / 10);
      System.out.print (n % 10 + " ");
}
```

Trace the output for the following method calls:

a. `recur4(1457);` 1457
b. `recur4(1881);` 1881
c. `recur4(-50);` 0

5. Consider the following recursive method:

```
public static void recur5 (int n){
      if (n > 1)
            recur5(n -1);
      System.out.print (n+ " ");
}
```

Trace the output for the following method calls:

a. `recur5(12);` 1 2 3 4 5 6 7 8 9 10 11 12
b. `recur5(8);` 1 2 3 4 5 6 7 8
c. `recur5(-50);` -50

