Exercise 1. Method Tracing – Recursion 1

```
public static void mystery1(int x, int y){
  if (y <= 0)
      System.out.print("0");
  else if(x > y)
  {
      System.out.print(x + " ");
      mystery1(x - y, y);
  }
  else
  {
      mystery1(x, y - x);
      System.out.print(y + " ");
   }
}
```

Write the values returned by each of the following calls:

mystery1(35, 14):

mystery1(5, 8):

Exercise 2. Method Tracing – Recursion 2

```
public static void mystery2(int n){
  if(n <= 1)
      System.out.print(“: “);
  else {
       System.out.print((n%2) + “ “);
       mystery2(n / 2);
       System.out.print(n + “ “);} }
```

Write the values returned by each of the following calls:

mystery2(25):

mystery2(38):

Exercise 3. Method Tracing – Recursion 3

```
public static int mystery3(int n){
  if(n < 0)
      return -mystery3(-n);
  else if(n < 10)
      return (n + 1) % 10;
  else
      return 100 * mystery3(n / 10) + (n + 1) % 10; }
```

What is the output of:

```
System.out.println(mystery3(576));


System.out.println(mystery3(-793));
```

Note: In all the subsequent methods, you are not allowed to use loops:

Exercise 4. Basic Recursive Methods

Numbers

Write a recursive method writeNums that accepts an integer parameter n and prints the first n integers starting with 1 in sequential order, separated by commas. *For example:*

writeNums(5); → 1, 2, 3, 4, 5

writeNums(12) → 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

Recursive Multiplication

Write a recursive method that takes two integers and returns their product. Note that the numbers can be positive or negative.

Recursive Power

Write a recursive method that takes two positive integers (x, y), and returns x raised to the power y. Use the property that:

When b is even: $x^b = (x^{b/2})^2$

When b is odd: $x^b = x * (x^{b/2})^2$

Exercise 5. Number of Digits

Write a recursive method that returns the number of digits in any integer. You are not allowed to use any string function or change the number into a string. Test your method of different numbers.

Exercise 6. Recursive Palindrome

Write a recursive method to check if a character array is a palindrome or not (have the same characters when read on reverse order). Test your method on different character arrays.

Exercise 7. Vowels to the End

Write a recursive method called vowelsToEnd that takes a string as a parameter and returns a string in which all of the vowels have been moved to the end. The vowels should appear in reverse order of what they were in the original word. For example, the call of vowelsToEnd("beauty") should return "btyuae".

Exercise 8. Numbers Sequence

Write a recursive method writeSequence that accepts an integer n as a parameter and prints a symmetric sequence of n numbers with descending integers ending in 1 followed by ascending integers beginning with 1, as shown below:

writeSequence(1); → 1
writeSequence(2); → 1 1
writeSequence(3); → 2 1 2
writeSequence(4); → 2 1 1 2
writeSequence(5); → 3 2 1 2 3
writeSequence(6); → 3 2 1 1 2 3
writeSequence(7); → 4 3 2 1 2 3 4
writeSequence(8); → 4 3 2 1 1 2 3 4
writeSequence(9); → 5 4 3 2 1 2 3 4 5
writeSequence(10); → 5 4 3 2 1 1 2 3 4 5

Notice that for odd numbers the sequence has a single 1 in the middle while for even values it has two 1s in the middle.