Problem 1.a:
\[ x = 6 \]
\[ y = 3 \]

Problem 1.b:
\[ a = 3 \]
\[ b = 4 \]
\[ d = 3.5 \]
\[ e = 3.0 \]

Problem 1.c:
\[ s = \text{“hellohello3”} \]
\[ t = \text{“hello3”} \]
\[ c = 3 \]

Problem 1.d, left side:
\[ 2.5 \]
\[ 4.5 \]

Problem 1.d, right side:
\[ 7 \]

Problem 1.e:
\[ 2 \]
\[ 1 \]
\[ 3 \]
\[ 2 \]
\[ 3 \]
\[ 3 \]

Problem 1.f:
\[ \text{“hello goodbye goodbye world”} \]
\[ \text{“hello goodbye goodbye goodbye goodbye world”} \]
\[ \text{“hello goodbye goodbye goodbye goodbye goodbye goodbye goodbye world”} \]

Problem 1.g:
\[ N = 4259 \]
\[ \text{sum} = 0 \]
\[ \text{sumStr} = \text{“”} \]
N = 425
sum = 9
sumStr = "9"

N = 42
sum = 14
sumStr = "95"

N = 4
sum = 16
sumStr = "952"

sum at the end = 20
sumStr at the end = "9524"

This loop adds up each digit in N, storing their sum in the variable sum. It also reverses the digits, in sumStr.

Problem 1.h:
a = [9, 11, 26, 19, 33, 49, 53, 59]

This code adds up the elements in the array.

Problem 1.i:
a = [2, 9, -7, 15, 16, 14, 6, 4]

This code swaps the order of pairs of elements in the array.
Problem 2.a

```java
System.out.println("Please enter a positive integer:");
int num = kb.nextInt();
System.out.println("The possible factors of " + num + " are:");

// I’m only writing it this way so it fits neatly
for(int possibleFactor = 1; possibleFactor <= num; possibleFactor++ )
{
    if(num % possibleFactor == 0) {
        System.out.println(possibleFactor);
    }
}
```

Problem 2.b (This was a bit longer than I intended, and longer than anything you’ll see on the exam)

```java
// first, read in 20 doubles from the keyboard, // and store them in an array:
System.out.println("please enter 20 doubles: ");
double [] someDoubles = new double[20];
for(int i=0; i<someDoubles.length; i++) {
    someDoubles[i] = kb.nextDouble();
}

// second, find the position of the smallest number
int smallestPosition = 0;   // position of smallest num
// value of smallest num
double smallestValue = Double.MAX_VALUE;

// Accumulation loop:
for(int i=0; i<someDoubles.length; i++) {
    if(someDoubles[i] < smallestValue) {
        smallestPosition = i;
        smallestValue = someDoubles[i];
    }
}

// swap the smallest number with the number at pos 0
double temp = someDoubles[0];
someDoubles[0] = smallestValue;
someDoubles[smallestPosition] = temp;
```
Problem 2.c

// generate 15 random numbers, each between 1 and 100
int [] numbers = new int[15];
for(int i=0; i<numbers.length; i++) {
    numbers[i] = rand.nextInt(100) + 1; // 1 to 100
}
Arrays.sort(numbers);
System.out.println("median = " + numbers[7]);
Problem 3

```java
import java.util.Scanner;

public class Steps {
    public static void main(String[] args) {
        Scanner kb = new Scanner(System.in);

        System.out.println("Enter number of steps");
        int numSteps = kb.nextInt();

        System.out.println("Enter width of steps");
        int width = kb.nextInt();

        System.out.println("Enter height of steps");
        int height = kb.nextInt();

        // loop that does 1 iteration for each step
        for(int step=0; step<numSteps; step++) {
            // loop that prints the right number of spaces
            // before the start of the top of the step
            for(int space=0; space<step*(width+1); space++) {
                System.out.print(" ");
            }

            // loop that prints the top of the step
            for(int hyphen = 0; hyphen < width; hyphen++) {
                System.out.print("-");
            }
            System.out.println();  // end of top of step

            // inner loop that prints
            // the right number of spaces for this row
            for( int space=0; space<(step+1)*(width+1)-1; space++) {
                System.out.print(" ");
            }
            System.out.println("|"); // end of 1 row of step
        }
    }
}
```