```
public class JasonFanPSet0 {
	public static void main(String[] args) {
	}
	//1
	public static int sumDouble(int a, int b) {
		if (a == b) {
			return 2 * (a + b);
		}
		else {
			return (a + b);
		}
	}
	//2
	public static boolean inRange(int a, int b) {
		return ((a >= 20 && a <= 30) && (b >= 20 && b <= 30)) || ((a >= 90 && a
<= 100) && (b >= 90 && b <= 100));
	}
	//3
	public static int absolute(int x) {
		if (x < 0) {
			return -1 * x;
		}
		else {
			return x;
		}
	}
	//4
	public static int closeNum(int a, int b) {
		if (absolute(10 - a) < absolute(10 - b)) {
			return a;
		}
		else if (absolute(10 - a) == absolute(10 - b)) {
			return 0;
		}
		else {
			return b;
		}
	}
	//5
	public static boolean isHarshad(int x) {
		int digitSum = (x % 10) + (x / 10);
		return x % digitSum == 0;
	}
	//6
	public static void leastToGreatest(int a, int b, int c) {
		if (a > b) {
			int temp = b;
			b = a;
			a = temp;
		}
		if (b > c) {
			int temp = c;
			c = b;
			b = temp;
		}
		if (a > b) {
			int temp = b;
			b = a;
			a = temp;
```

```
		}
		System.out.println("Ascending order: " + a + ", " + b + ", " + c);
	}
	//7
	public static int roundNum(double x) {
		if (x > 0) {
			x = (int) (x + .5);
		}
		else {
			x = (int) (x - .5);
		}
		return (int) x;
	}
	//8
	public static boolean twinTrouble(boolean aSmile, boolean bSmile) {
		if (aSmile == bSmile) {
			return true;
		}
		else {
			return false;
		}
	}
	//9
	public static void quadRoots(int a, int b, int c) {
		if (Math.pow(b, 2) - (4 * a * c) < 0) {
			System.out.println("No Real Roots!");
			return;
		}
		double root1 = ((-1 * b) + Math.sqrt(Math.pow(b, 2) - (4 * a * c))) /
(2 * a);
		double root2 = ((-1 * b) - Math.sqrt(Math.pow(b, 2) - (4 * a * c))) /
(2 * a);
		if (root1 == root2) {
			System.out.println("Double root: x = " + root1);
		}
		else {
			System.out.println("Root1: x = " + root1 + " " + "Root2: x = " +
root2);
		}
	}
	//10
	public static double calcForce(int m1, int m2, int r) {
		return (6.67 * (Math.pow(10, -11)) * m1 * m2) / (Math.pow(r, 2));
	}
	//11
	public static double barycenter(double m1, double m2, double a) {
		return (a * m2) / (m1 + m2);
	}
	//12
	public static double yPosition(double vel, double t) {
		final double g = -9.8;
		return (.5 * g * Math.pow(t, 2)) + (vel);
	}
	//13
	public static double triArea(int a, int b, int c) {
		double s = (double) (a + b + c) / 2;
		return Math.sqrt(s * (s - a) * (s - b) * (s - c));
	}
	// 14
```

```
      public static double calcLJ(int eps, int sig, int r) {
            return (((double)48 * eps) / Math.pow(r, 2)) * (Math.pow((double)sig /
r, 12) - (.5 * Math.pow((double)sig / r, 6)));
      }
}
```