Question 1: Rename the following method to better describe what it does, and simplify it by replacing the three arithmetic operators with one operator.

```java
public static int strange(int x, int y) {
    return x-x/y*y;
}
```

Question 2: Complete the following method, which returns the largest integer that is less than or equal to the specified value. Ensure that your method works for both negative and positive values of `x`.

```java
public static int floor(double x) {
}
```

Question 3: Next to each statement in the following program fragment, write the value of `n`.

```java
int n = 3;
--n;
++n;
```

```java
n += 3;
```

```java
n /= 3;
```

```java
n %= 3;
```
Question 4 .............................................................. (5 points)
   How many bits in a Java byte?
   How many bytes in a Java float?
   How many bytes in a Java double?
   How many bytes in a Java short?
   How many bytes in a Java int?

Question 5 .............................................................. (10 points)
   Complete the following method, which returns true if the specified array is sorted with all elements in increasing order; false, otherwise.

   public static boolean isSorted(float[] x) {
   
   }

Question 6 .............................................................. (10 points)
   Complete the following method, which zips the two specified arrays together and returns a new array with elements \{x[0], y[0], x[1], y[1], \ldots\}. You may assume that the two arrays have equal lengths.

   public static float[] zip(float[] x, float[] y) {
   
   }
Question 7 ................................................................. (10 points)

Complete the following method, which returns a new 2D array with specified dimensions \( n1 \) and \( n2 \), and all elements copied from the specified 1D array \( x \). Assume that the 1D array \( x \) has length \( n1 \times n2 \).

```java
public static float[][] reshape(float[] x, int n1, int n2) {
```

```java
}
```

Question 8 ................................................................. (15 points)

Complete the following method, which reads a binary file containing (1) two ints representing the width and height of an image, followed by (2) a sequence of floats representing the image pixels that are returned as a 2D array.

```java
public static float[][] readBinary(String fileName) {
```

```java
}
```
Question 9... (20 points)

Complete the following class, which mimics a stopwatch. Use the standard method `System.nanoTime()`, which returns a `long`, the number of nanoseconds that have elapsed since some fixed but otherwise arbitrary system time.

```java
public class Stopwatch {
    /** Returns the stopwatch time, in seconds. Does not start or stop. */
    public double time() {
    }

    /** Starts this stopwatch. If the stopwatch is running, does nothing. */
    public void start() {
    }

    /** Stops this stopwatch. If the stopwatch is not running, does nothing. */
    public void stop() {
    }

    /** Stops this stopwatch and resets its time to zero. */
    public void reset() {
    }
}
```
Question 10 ................................................................. (15 points)

Using the class `Stopwatch` defined on the previous page, write a complete Java program that prints the number of seconds required to compute the sum $1^3 + 2^3 + 3^3 + \cdots + n^3$, for $n = 1000$. 