Methods

For exercises 1, 2 and 3, you only have to write the method header, NOT the actual method.

1. Write a method header for a method named `translate` that takes an integer parameter and returns a double.
        static double translate(int x)

2. Write a method header for a method names `find` that takes a `String` and a `double` as parameters and returns an integer.
        static int find(String x, double y)

3. Write a method header for a method named `printAnswer` that takes three doubles as parameters and does not return anything.
        static void printAnswer(double x, double y, double z)

For each of the following questions, **write a method** that performs the specified task, then **create the main method** (in the same class) to test your method.

4. **Welcome.java**   Write a method for the following header.
        public static String welcomeMessage (String name, int
        visitorNum)
    The method should return a welcome message that includes the user's name and visitor number. For example, if the parameter were "Joe" and 5, the return string would be "Welcome Joe! You are visitor number 5."

5. **SoundAlarm.java**   Write a method called `alarm` that prints the string "Alarm!" several times on separate lines. The method should accept an integer parameter that tells it how many times that string is printed. Print an error message if the parameter is less than 1.

    HINT: This method will *print* lines, it does not return a String. The return type should be void.

6. **CalculateSum.java**   Write a method called `sumRange` that accepts two integer parameters that represent a range such as 50 to 75. Issue an error message and return zero if the second parameter is less than the first. Otherwise, the method should return the sum of the integers in that range (inclusive).

7. **CheckDivisibility.java** Write a method called `evenlyDivisible` that accepts two integer parameters and returns true if the first parameter can be

evenly divided by the second, or vice versa, and false if it can't be. Return false if either parameter is zero.