Java Assignment 1

This assignment is about parallelizing a quite hard computation, namely the factorization of a possibly relatively large integer. You should implement a program called ‘Factorizer’ according to the specification below.

The program should take two integers as input:

- the first integer should be the product of two prime numbers (internally represented as a BigInteger);
- the second integer should be the number of concurrent threads the program should use for the computation.

The program should output the following:

- the two factors of the inputted integer (if there are such factors);
- the computation time.

Note! You should not implement any complicated algorithm for finding prime factors. You should only use the simple approach described by the following ‘run’-method:

```
public void run() {
	BigInteger number = min;
	while (number.compareTo(max) <= 0) {
		if (product.remainder(number).compareTo(BigInteger.ZERO) == 0) {
			factor1 = number;
			factor2 = product.divide(factor1);
			return;
		}
		number = number.add(step);
	}
}
```

In the code above ‘product’ is the inputted product, ‘min’ is the least value to investigate, ‘max’ is the greatest value to investigate, and ‘step’ is the number you add to get the next number to investigate; and ‘factor1’ and ‘factor2’ are the resulting factors of ‘product’.

Note! You might need to modify the code above for it to work properly for your solution.

The requirements for dealing with incorrect input are as follows:
If the inputted product value is not a product of two prime numbers then it is either a prime number itself or a product of more than two prime numbers. If it is a prime number itself then the program should output “No factorization possible”. If it is a product of more than two prime numbers then the program should still output two factors, which do not have to be prime numbers. In other words, no further factorization after finding two factors needs to be done.

Other implementation requirements:
You are not allowed to use any predefined classes or key words for managing concurrency that have not been presented on the two first lectures on the course.

Basic task (grade C/D/E/F)

Your task is to implement a concurrent and thread-safe program according to this specification, and to make the program as performant as possible **except** for the special requirement described under "Advanced task" below.

Advanced task (grade A/B/C/D/E/F)

Your task is to implement a concurrent and thread-safe program according to the specification above, and to make the program as performant as possible **including** the requirement that the execution of all threads should be properly terminated when the two factors have been found in one thread. The termination should be done as soon as reasonable to achieve the best overall performance.

To be able to get grade A or B you should submit your solution before the first deadline.

Submission

Your submission should be a **single file** called **Factorizer.java**, and it should be submitted to the appropriate place in iLearn2.

The first line in the file should include the name of the author.

Good luck!