edu.gmu.cs.swe619.assignments.assignment02

Class FindPrimeFactor

java.lang.Object
     edu.gmu.cs.swe619.assignments.assignment02.FindPrimeFactor

---

```
public class FindPrimeFactor
extends java.lang.Object
```

Method Summary

All Methods	Static Methods	Concrete Methods

Modifier and Type	Method and Description
static int	**findPrimeFactor**(java.util.List<java.lang.Integer> a, java.util.List<java.lang.Integer> b) Finds the least index at which b[i] is a prime factor of a[i].
static boolean	**isPrime**(java.lang.Integer x) Determines whether the given integer is prime or not.

Methods inherited from class java.lang.Object

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Method Detail

findPrimeFactor

```
public static int findPrimeFactor(java.util.List<java.lang.Integer> a,
                                  java.util.List<java.lang.Integer> b)
```

Finds the least index at which b[i] is a prime factor of a[i].

Note that a[] and b[] need not be of the same length.

Example 1: findPrimeFactor ([12, 25, 18, 8], [6, 2, 3, 2]) = 2. Note that 6 is a factor of 12, but is not prime, and 2 is prime, but is not a factor of 25. However, 3 is a prime factor of 18. Hence, index "2" is the correct answer. index "3" is not a possible answer, because the third index is not the least index with the desired property.

Example 2: findPrimeFactor([12, 15, 24], [4, 2, 1]) results in IllegalStateException. In this case, 12 is divisible by 4, but it is not prime. 15 is not divisible by 2, and 1 is not prime.

Parameters:

a - list of integers, one of which must have a prime factor that matches the element at the same index in list b

b - list of integers, one of which must be prime and a factor of the element in list a at the same index

Returns:

the least index i, for which b[i] is a prime factor of a[i]

Throws:

java.lang.NullPointerException - if either input list (a or b) is null

java.lang.IllegalStateException - if there is no index i in the given lists for which b[i] is a prime factor of a[i]

isPrime

```
public static boolean isPrime(java.lang.Integer x)
```

Determines whether the given integer is prime or not.

This method uses the standard rules for computing primality. 0, 1, and all negative numbers are considered to be non-prime.

Parameters:

x - the number to test for primality

Returns:

whether x is prime (true / false)

Throws:

java.lang.NullPointerException - if x is null