

総合演習課題

## 作成プログラム概要

**N** 桁の数字を受け取り、素因数分解した結果のリストを返す **Program** を作成せよ

使用言語：Java（version は問わない）

呼び出し方：Factorization.factorize(int)

仕様

製品コード

以下の出力になるように作成してください

Input : int	Output :List[int]
10	[2,5]
47	[47]
18	[2,3,3]

テストコード

テストデータの txt を読み込み、一致しているかをチェックするプログラムを作成してください  
なお、テストデータのサンプルとして以下を参照する [Test1](#)

Input1 : List[List[int]](Textから読みとった)	Input2 : List[List[int]](作成したメソッドで算出)	Output :float（割合）
[[2,5],[2,2,3]]	[[2,5],[2,2,3]]	1.0
[[2,5],[2,2,3]]	[[3,5],[2,2,3]]	0.5

データ形式

```
input:[output[1],output[2],output[3] ... ,output[n]]
```

例

```
68:2,2,17
```

## ルール

誰かが代表してリポジトリを作成してください  
全員、Branch を切って作業してください。

チーム内を以下のように分ける

- A チーム：製品プログラム作製（2）
- B チーム：テストプログラム作製（1+1）

PR は書いて、別チームにレビューしてもらってください。

最後に性能評価します。

確認するのに楽になるので製品コードは 1 つにまとめてください。

## 仕様変更

### 製品の出力変更

Input : int	Output :List[string]
18	[2,3^2]

### テストコード

テストデータの txt を読み込み、一致しているかをチェックするプログラムを作成してください  
なお、テストデータのサンプルとして以下を参照する [Test2](#)

Input1 : List[List[string]](Textから読みとった)	Input2 : List[List[string]](作成したメソッドで算出)	Output :float（割合）
[[2,5],[2^2,3]]	[[2,5],[2^2,3]]	1.0
[[2,5],[2^2,3]]	[[2,5],[2^3,3]]	0.5

以上



振り返り

## 今回、チャレンジしていただいた観点

- 2チームにわかれてペアプロを行う ← ペアモブプロの振り返り
- 2チームが GitHub 上で PR をだす。← Git, GitHub 研修
- コードレビューを双方に行う ← リーダブルコード振り返り
- 要求変更を通して設計の大切さを考える
- テスト駆動開発を少しだけ体験してほしい
- ちょっと数学やる ← リーダブルコード研修の続き

## 要求変更を通して設計の大切さを考える

Agile 開発では少しずつ物を作っていくことが常です。

ので最初に決めた方針（設計）が後々の機能追加のコストに大きく影響します。

現場でさまざまな経験をしてください！

## テスト駆動開発を少しだけ体験してほしい

品質を高めるのは、テストではなく設計とコーディングです。  
けど、それを測定するのはテストです。

テストを書いているだけでは品質は上がらない（体重計に乗っただけでは痩せない）  
可視化しておくことが大事です。

この Agile 研修ではそこまでできないが（テストコードを書いている時間はない）  
現場に出る前に知識とだけでも知っていて欲しかったの入れました！

## ちょっと数学やる

プログラマとしての頭の使い方を鍛えるためにも数学は必須だと僕は思います。  
Track でも AtCoder でもなんでもいいので、研鑽は続けてください！

参考 URL

[CodeWars](#)

[AtCoder](#)

現場で待ってます！

**Dx** 研修卒業生一同