

## 初心者の為の練習問題

原文は[Scala exercises for beginners](#)を参照の事。関数型言語の初心者向けの良い課題だと思うのだが、...が、関数型言語に不慣れだと、そもそもどんな関数を作る事が期待されているのか判らないかもしれないなあ。(メソッド名とか関数の型から大体的見当がつかない?)

ソースコード中の error("課題") の部分を自分の書いたコードで置き換える事が期待されています。

初心者の回答を自動で採点する為に、scalacheck で回答の正当性を検証する為の方法を誰か解説しない?

```
// 下記の List のメソッドは使用してはならない:  
// * length  
// * map  
// * filter  
// * ::: (および ++ のようなその変形)  
// * flatten  
// * flatMap  
// * reverse (および reverseMap, reverse_::: のような変形)  
// これはまた、List に対する for-構文の使用も禁止している。  
// 自分で書いた関数は使用して良い、例えば問題 2 では問題 1 あるいは問題 3 を使用して良い。  
// 許可された既に存在するメソッドを適切に使用した場合は、エレガントさが評価される。
```

```
// 満点: 66点
```

```
object Exercises {  
  def succ(n: Int) = n + 1  
  def pred(n: Int) = n - 1
```

```
// Exercise 1 (問題 1)  
// Relative Difficulty 1 (難易度: 1)  
// Correctness: 2.0 (正しい回答に: 2.0 点)  
// Performance: 0.5 (性能: 0.5 点)  
// Elegance: 0.5 (エレガントさ: 0.5 点)  
// Total: 3 (合計: 3)
```

```
def add(x: Int, y: Int): Int = error("課題: x, yは 0 または正の数と仮定せよ。Int に対する +, - の使用を禁止する。上述の s
```

```
// Exercise 2  
// Relative Difficulty: 2  
// Correctness: 2.5 marks  
// Performance: 1 mark  
// Elegance: 0.5 marks  
// Total: 4  
def sum(x: List[Int]): Int = error("課題")
```

```
// Exercise 3  
// Relative Difficulty: 2  
// Correctness: 2.5 marks  
// Performance: 1 mark  
// Elegance: 0.5 marks  
// Total: 4  
def length[A](x: List[A]): Int = error("課題")
```

```
// Exercise 4  
// Relative Difficulty: 5  
// Correctness: 4.5 marks  
// Performance: 1.0 mark  
// Elegance: 1.5 marks  
// Total: 7  
def map[A, B](x: List[A], f: A => B): List[B] = error("課題")
```

```
// Exercise 5  
// Relative Difficulty: 5
```

```
// Correctness: 4.5 marks
// Performance: 1.5 marks
// Elegance: 1 mark
// Total: 7
def filter[A](x: List[A], f: A => Boolean): List[A] = error("課題")
```

```
// Exercise 6
// Relative Difficulty: 5
// Correctness: 4.5 marks
// Performance: 1.5 marks
// Elegance: 1 mark
// Total: 7
def append[A](x: List[A], y: List[A]): List[A] = error("課題")
```

```
// Exercise 7
// Relative Difficulty: 5
// Correctness: 4.5 marks
// Performance: 1.5 marks
// Elegance: 1 mark
// Total: 7
def concat[A](x: List[List[A]]): List[A] = error("課題")
```

```
// Exercise 8
// Relative Difficulty: 7
// Correctness: 5.0 marks
// Performance: 1.5 marks
// Elegance: 1.5 mark
// Total: 8
def concatMap[A, B](x: List[A], f: A => List[B]): List[B] = error("課題")
```

```
// Exercise 9
// Relative Difficulty: 8
// Correctness: 3.5 marks
// Performance: 3.0 marks
// Elegance: 2.5 marks
// Total: 9
def maximum(x: List[Int]): Int = error("課題")
```

```
// Exercise 10
// Relative Difficulty: 10
// Correctness: 5.0 marks
// Performance: 2.5 marks
// Elegance: 2.5 marks
// Total: 10
def reverse[A](x: List[A]): List[A] = error("課題")
}
```

- 答え合わせってどうすれば? --\_kzt\_ (2010-01-09 12:08:58)
- あ、答えってというのは変ですね、回答例かな。 --\_kzt\_ (2010-01-09 12:12:24)

名前:	<input type="text"/>
コメント:	<input type="text"/>

投稿