

Problema 2: Validar números de cartões de crédito



Como é que os sites de compras *online* rejeitam rapidamente erros na digitação dos números dos cartões de crédito? Não é necessário pesquisar numa gigantesca base de dados: há um método simples para distinguir números corretos de uma qualquer sequência aleatória de algarismos usando uma soma de verificação.

Neste exercício pretende-se implementar o algoritmo de validação descrito pelos passos seguintes.

1. Decomponha o número em algarismos. Por exemplo¹, se o número for 1386, a lista de algarismos é [1,3,8,6].
2. Duplique o valor de cada segundo algarismo começando pela direita, ou seja: o último algarismo fica inalterado; o penúltimo algarismo é duplicado; o antepenúltimo algarismos fica inalterado; e assim sucessivamente. Por exemplo, para a lista de algarismos [1,3,8,6] obtemos [2,3,16,6].
3. Some os algarismos de todos os valores da lista resultante (que podem ter um ou dois algarismos). Para a lista [2,3,16,6] obtemos $2 + 3 + 1 + 6 + 6 = 18$.
4. Calcule o resto de divisão da soma por 10. O número original será válido se o resultado for 0. Para o exemplo anterior, o resto será $18 \bmod 10 = 8$ (ou seja, não é um número válido).

Tarefa

Pretende-se que traduza o algoritmo acima como uma função

```
validar :: Integer -> Bool
```

que testa se um inteiro pode ser um número de cartão de crédito válido. Sugestão: defina funções auxiliares para cada um dos passos acima.

Exemplos

```
validar 4012888888881881 = True
```

```
validar 4012888888881882 = False
```

Adaptado de um exercício de Doaitse Swierstra, Universidade of Utrecht, 2008-2009.

¹Os números de cartão de crédito têm tipicamente mais algarismos; usamos um exemplo curto apenas para simplificar as contas.