

## Exercice 1

La fonction `isGoodEnough` présentée au cours n'est pas très efficace pour trouver les racines carrées de très petits nombres. D'autre part, dans les ordinateurs réels, les opérations arithmétiques travaillant sur les nombres à virgule flottante sont presque systématiquement effectuées avec une précision limitée. Cela rend notre test inadéquat pour de très grands nombres.

Expliquez ces deux affirmations en décrivant le genre de problèmes rencontrés par notre fonction de test avec des nombres très petits et très grands.

Une stratégie alternative pour réaliser la fonction `isGoodEnough` consiste à examiner comment `guess` change d'une itération à l'autre et à s'arrêter lorsque ce changement est une petite fraction de `guess`. Écrivez une fonction de calcul de racine carrée qui utilise cette technique.

## Exercice 2

La méthode de Newton pour les racines cubiques se base sur le fait que si  $y$  est une approximation de la racine cubique de  $x$ , alors une meilleure approximation est donnée par la formule

$$\frac{x/y^2 + 2y}{3}$$

Utilisez cette formule pour écrire une fonction de calcul de racine cubique analogue à la fonction de calcul de racine carrée vue au cours.

## Exercice 3

Le triangle ci-dessous est nommé *triangle de Pascal*.

$$\begin{array}{cccccccc}
 & & & & & & & 1 \\
 & & & & & & 1 & 1 \\
 & & & & 1 & 2 & 1 & \\
 & & 1 & 3 & 3 & 1 & & \\
 & 1 & 4 & 6 & 4 & 1 & & \\
 1 & 5 & 10 & 10 & 5 & 1 & & 
 \end{array}$$

Les nombres sur les bords du triangle valent tous 1, tandis que les nombres à l'intérieur sont obtenus en sommant les deux nombres au dessus. Écrivez une fonction récursive qui calcule les éléments d'un tel triangle.

## Exercice 4

Écrivez une fonction récursive qui vérifie l'équilibre des parenthèses dans une chaîne de caractères donnée. Par exemple, la fonction doit retourner `true` pour les chaînes

```
"(if (zero? x) max (/ 1 x))"
```

et

```
"Je lui ai dit (que ce n'est pas (encore) fini).  
(Mais il faisait la sourde oreille)"
```

Elle doit retourner `false` pour `" :-)"` et `"() )"`. Le dernier exemple montre qu'il ne suffit pas de vérifier qu'une chaîne contient le même nombre de parenthèses ouvrantes que de parenthèses fermantes.