

# Puissance nième d'un nombre [rc02] - Exercice

Karine Zampieri, Stéphane Rivière

Unisciel

sciel

algotprog

UNIVERSITÉ  
HAUTE-ALSACE

Version 21 mai 2018

## Table des matières

<b>1</b>	<b>Puissance nième d'un nombre / pgpuissance</b>	<b>2</b>
1.1	Puissance naïve . . . . .	2
1.2	Stratégie basée sur la parité . . . . .	2
1.3	Autre stratégie basée sur la parité . . . . .	2
1.4	Fonction itérative . . . . .	3
1.5	Programme de test . . . . .	3

## Python - Puissance nième d'un nombre (TP)



**Mots-Clés** Récursivité des actions ■

**Requis** Schéma itératif ■

**Difficulté** ●●○



### Objectif

Cet exercice calcule récursivement la puissance  $x^n$  d'un réel  $x$  par un entier  $n \geq 0$  de plusieurs manières. Dans le même ordre d'idées, l'exercice @[Fonction produit] calcule récursivement le produit  $a \cdot n$  d'un réel  $a$  par un entier  $n \geq 0$  de plusieurs manières.

# 1 Puissance nième d'un nombre / pgpuissance

## 1.1 Puissance naïve

Soient un réel  $x$  et un entier  $n \geq 0$ .

L'idée la plus simple pour le calcul de  $x^n$  consiste à utiliser :

$$x^n = \begin{cases} 1 & \text{si } n = 0 \\ x \cdot x^{n-1} & \text{sinon} \end{cases}$$



Écrivez une fonction récursive `puiss1(x,n)` qui calcule et renvoie la puissance d'un réel  $x$  (avec  $n$  entier positif) à partir de la définition par récurrence.



Combien y a-t-il d'appels récurrents ?

## 1.2 Stratégie basée sur la parité

La propriété suivante accélère le calcul :

$$x^n = \begin{cases} 1 & \text{si } n = 0 \\ (x^{n \div 2})^2 & \text{si } n \text{ pair} \\ x \cdot x^{n-1} & \text{sinon} \end{cases}$$



Écrivez une fonction `carre(x)` qui renvoie le carré de  $x$  (réel).



Écrivez une fonction récursive `puiss2(x,n)` qui calcule et renvoie la puissance d'un réel  $x$  (avec  $n$  entier positif) comme décrit ci-avant.



## 1.3 Autre stratégie basée sur la parité

Une autre façon d'accélérer significativement le calcul de la puissance (en le ramenant à au plus  $2 \log_2 n$ ) est la propriété suivante :

$$x^n = \begin{cases} 1 & \text{si } n = 0 \\ (x^2)^{n \div 2} & \text{si } n \text{ pair} \\ x \cdot x^{n-1} & \text{sinon} \end{cases}$$

Par exemple, on calcule  $x^{10}$  en quatre multiplications au lieu de 9 :

$$x^{10} = (x^2)^5 = x^2 \left( (x^2)^4 \right) = x^2 ((x^2)^2)^2$$



Écrivez une fonction récursive `puiss3(x,n)` qui calcule et renvoie la puissance d'un réel `x` (avec `n` entier positif) en appliquant la relation ci-dessus.



Donnez la suite des transformations de `(x,n,y)` pour le calcul de  $5^8$  puis le calcul de  $5^7$ . Concluez.

## 1.4 Fonction itérative

La fonction du problème précédent étant récursive terminale,

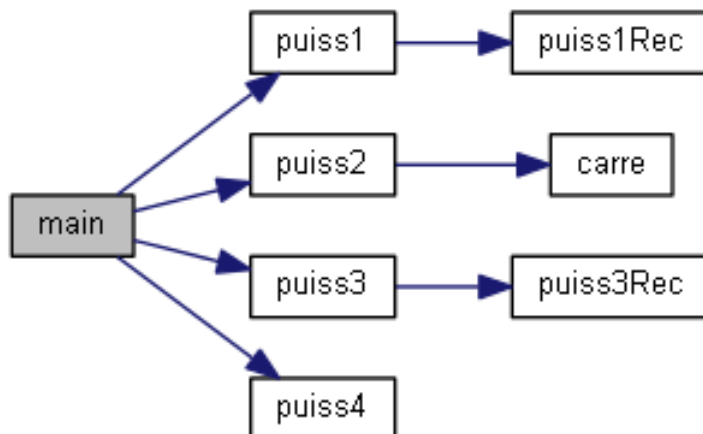


Écrivez une fonction itérative `puiss4(x,n)`, équivalente à la version récursive terminale `puiss3(x,n)`, en remplaçant la liste des paramètres des appels récursifs par des affectations appropriées.

## 1.5 Programme de test



Écrivez un script qui saisit un réel et un entier puis calcule et affiche le résultat de chacune des fonctions.



Testez. Exemple d'exécution :

```
Puissance x? 5
Ordre n? 7
==> puiss1(x,n) vaut 78125
==> puiss2(x,n) vaut 78125
==> puiss3(x,n) vaut 78125
==> puiss4(x,n) vaut 78125
```