Devoir maison 1

pour le mardi 4 novembre 2014

Consignes. *Si ce n'est pas encore fait installez Python sur votre machine ; les réponses aux questions doivent être rédigées* ***sur une copie****. Testez vos programmes en machine.*

Exercice 1. Home-made square root

Q1. Expliquer le fonctionnement du programme Python `racine2.py` ci-dessous.

```
1 from math import sqrt
2 nbre=input("Donnez un nombre postif : ")
3 a=float(nbre)
4 if a>=0 :
5     print("La racine carrée de ",a,"vaut approximativement ",sqrt(a))
6 else :
7     print("Ceci n'est pas un nombre positif.")
```

On se propose de concevoir un programme Python calculant une racine carrée **sans utiliser** la fonction `sqrt`. Pour extraire la racine carrée d'un nombre positif a on utilise la méthode suivante :

- on part d'une valeur $x_0 \geq \sqrt{a}$;
- on calcule les termes de la suite définie par la relation de récurrence :

$$x_{k+1} = \frac{1}{2}\left(x_k + \frac{a}{x_k}\right);$$

- on justifie que la suite $(x_k)_{k\geq 1}$ est décroissante et a pour limite $\sqrt{a}$. Les termes x_k pour des valeurs suffisamment grandes de k donnent une approximation de $\sqrt{a}$.

La convergence de la suite est *quadratique* : notant $\delta_k = u_k - \sqrt{a}$ on vérifie que pour $k \geq 1$: $\delta_{k+1} \leq \dfrac{\delta_k^2}{2\sqrt{a}}$. Cela signifie donc que si $\delta_k \leq 10^{-p}$ alors $\delta_{k+1} \leq \dfrac{10^{-2p}}{2\sqrt{a}}$. On considère donc que le nombre de décimales exactes dans l'approximation de $\sqrt{a}$ par x_k double à chaque itération (pour $a \geq 1$).

Q.2 Puisque tout nombre décimal peut s'écrire sous la forme $N.10^{2k}$ avec N un entier naturel et k un entier relatif, il est suffisant de savoir calculer des racines carrées d'entiers. On se propose de concevoir une fonction `racine2ent(N)` dont l'argument d'entrée `N` est de type `int`. La fonction retourne la partie entière de la racine carrée de N. La fonction met en œuvre la méthode décrite ci-dessus pour le nombre $a = N$.

(a) Proposer une méthode simple permettant de choisir un terme initial x_0 relativement proche de $\sqrt{N}$. À défaut on prend $x_0 = N$.

(b) Rédiger la fonction `racine2ent(N)`. Elle s'appuie sur un calcul itératif des termes x_k qui prend fin lorsque $\lfloor x_k \rfloor^2 \leq N$ où $\lfloor x_k \rfloor$ désigne la partie entière de x_k. Expliquer pourquoi l'algorithme termine. Tester votre fonction pour les valeurs $N = 10^{10}, 10^{20}, 10^{40}$ et 10^{60}. Estimer le nombre d'itérations ? Quel(s) constat(s) peut-on faire ?

Remarque. Il serait préférable d'utiliser une méthode qui ne fait pas appel au calcul sur les nombres de type `float` : pour cela, on peut substituer les divisions dans la relation de récurrence par des **divisions entières**. Les questions de terminaison et correction seraient alors à nouveau posées : on admet que c'est encore correct.

Q3. À l'aide des deux questions précédentes concevoir un programme Python qui demande à l'utilisateur un nombre entier N ainsi qu'un entier p. Le programme affiche la valeur de $\sqrt{N}$ avec au moins p décimales justes.
L'éxécution du programme donne dans l'interpréteur :

```
>>>
Donnez un nombre entier positif : 2
Nombre de décimales pour la racine carrée ? 99
La racine carrée de 2 vaut approximativement :
1.41421356237309504880168872420969807856967187537694807317667973799073247846210703885038753432764157 2
```

Exercice 10. Conversions décimal-binaire

« Tu vois, le monde se divise en 10 catégories, ceux qui connaissent le binaire et ceux qui creusent. »

Décomposition binaire
Pour tout entier naturel $N \neq 0$ il existe un unique entier $n_0 \geq 0$ et une unique famille $(x_k)_{0 \leq k \leq n_0}$ constituée de 0 ou 1 avec $x_{n_0} \neq 0$ tels que :

$$N = x_{n_0}.2^{n_0} + x_{n_0-1}.2^{n_0-1} + \cdots + x_k.2^k + \cdots + x_1.2^1 + x_0.2^0$$

Notation. *L'écriture binaire* du nombre N est $x_{n_0}x_{n_0-1}\cdots x_k\cdots x_1x_0$.

Exemple. $13 = 1.2^3 + 1.2^2 + 0.2^1 + 1.2^0$. Donc le nombre 13 (en base dix) s'écrit en binaire 1101.

Q1. Écrire une fonction Python `conversionD(N)` qui retourne l'écriture décimale d'un nombre N donné en entrée sous forme binaire. L'entrée N pourra être convertie en chaîne de caractères à l'aide de la fonction `str()`. On n'utilise pas de fonction Python préexistante qui réaliserait cette conversion.
On peut mettre en œuvre la démarche suivante. Par exemple, pour $N = 1001$ l'écriture décimale s'obtient par l'opération :

$$1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 9$$

Sur la chaîne de caractères `N='1001'` cela s'obtient par le calcul :

```
N[0]*2**3 + N[1]*2**2 + N[2]*2**1 + N[3]*2**0
```

Rappel : la fonction `len` permet d'obtenir la longueur d'une chaîne de caractères.

Dans l'autre sens, pour trouver l'écriture binaire d'un entier on peut procéder en utilisant un algorithme que nous présentons sur l'exemple $N = 37$.

- On cherche la plus grande puissance de 2 inférieure ou égale à 37. C'est $2^5 = 32$.
- On itère le procédé avec le nombre $37 - 2^5 = 5$. C'est $2^2 = 4$...
- Lorsqu'on arrive à 0 nous obtenons la décomposition :

$$37 = 1.2^5 + 0.2^4 + 0.2^3 + 1.2^2 + 0.2^1 + 1.2^0$$

L'écriture binaire de 37 c'est 100101.

Objectif. Écrire un programme Python `conversion2.py` qui demande à l'utilisateur un entier non nul N écrit en base 10. Le programme affiche l'écriture binaire du nombre N. On n'utilise pas de fonction Python préexistante qui réaliserait cette conversion.

Q10. Rédiger une fonction Python `exp2max(N)` qui retourne la plus grand exposant r tel que $2^r \leq N$.
Exemple : `exp2max(37)` retourne `5`.
Estimer la complexité en nombre de multiplications d'entiers de votre fonction.

Q11. On donne un entier N. Rédiger des instructions Python permettant de construire, à partir d'une liste vide, la liste des exposants successifs obtenus par l'algorithme de décomposition binaire appliqué à N.
Exemple : pour `N=37` on doit obtenir la liste `[5,2,0]`.

Q100. Écrire une fonction `transfo(L)` qui convertit la liste des puissances obtenue précédemment en une chaîne des caractères `0` et `1` correspondant à l'écriture binaire du nombre. Expliquer la démarche mise en œuvre.
Exemple : `transfo([5,2,0])` retourne la chaîne de caractères `100101`.

Concevoir le programme attendu (on ne demande pas de le rédiger sur la copie).