DEVOIR SURVEILLÉ 1

1 Algorithmes de calcul de racines carrées

Dans des temps pas si lointains, les étudiants n'avaient pas de calculatrice mais ils savaient calculer des racines carrées à la main...

A- La méthode grecque. Les Babyloniens savaient déjà calculer des valeurs approchées de racines carrées. On trouve par exemple sur la tablette YBC 7289 une approximation de $\sqrt{2}$ à 10^{-5} près, ce qui est déjà remarquable. L'idée qui mène à ce résultat a été théorisée par le mathématicien grec Héron d'Alexandrie au I^e siècle. Elle est purement géométrique : pour calculer le coté d'un carré d'aire a (c'est-à-dire la racine de a), on part d'un rectangle de coté arbitraire x que l'on transforme en un autre rectangle de même aire, mais à la forme « un peu plus carrée », en prenant pour l'un de ses cotés la moyenne arithmétique des deux cotés du rectangle précédent, c'est-à-dire $\frac{1}{2}\left(x+\frac{a}{x}\right)$. La méthode, qui prendra par la suite le nom de *méthode de Héron*, aboutit à considérer $\sqrt{a}$ comme la limite de la suite récurrente de premier terme $x_0 > 0$ (on peut prendre par exemple $x_0 = a$) et vérifiant $\forall n \in \mathbb{N}, x_{n+1} = \frac{1}{2}\left(x_n + \frac{a}{x_n}\right)$. On démontrera en mathématiques que cette suite converge bien vers $\sqrt{a}$ et que le nombre de décimales exactes va doubler à chaque itération.

1. Écrire une fonction `suite_heron` prenant en argument un nombre a et un entier n, et renvoyant x_n, où x est la suite définie ci-dessus. Cette fonction devra être itérative (avec une boucle `for`).
2. Écrire une variante récursive de la fonction précédente. On fera attention à garder un coût en temps linéaire.
3. Écrire une fonction `racine_heron` prenant en argument un nombre a et renvoyant une approximation de sa racine carrée en renvoyant x_6.

B- La méthode « de la potence ». Cette méthode est celle qui était enseignée aux écoliers pour calculer les racines carrées à la main avant l'ère de la calculatrice. Elle s'organise un peu comme une division euclidienne, le long d'une « potence » en angle droit.
On commence par séparer le nombre en tranches de deux chiffres à partir de la droite ; il se peut que la tranche la plus à gauche ne comporte qu'un chiffre. En effet, la partie entière de la racine carrée d'un nombre contient autant de chiffres qu'il y a de tranches de deux chiffres dans le nombre. Par exemple, si on souhaite calculer la racine carrée de 76176, on écrira 7|61|76.

Pour trouver le premier chiffre de la racine, on prend la première tranche, ici 7, et on cherche le plus grand carré contenu dans 7 qui est $4 = 2^2$. Le premier chiffre de la racine est donc 2 et on retranche 4 de 7, il reste 3.

7	6	1	7	6	2
- 4					
3					

Ensuite, on descend la tranche suivante pour obtenir 361, on double la racine que l'on a obtenue, donc ici $2 \times 2 = 4$ et on cherche le plus grand chiffre entre 0 et 9 tel que 4_ × _ soit inférieur ou égal à 361.

7	6	1	7	6	2
- 4					4_ × _
3	6	1			

On a $49 \times 9 = 441$ qui est trop grand, $48 \times 8 = 384$ est trop grand également, et $47 \times 7 = 329$ convient. 7 est donc le deuxième chiffre de la racine et on soustrait 329 à 361 ce qui donne 32.

7	6	1	7	6	2	7
- 4					47	× 7 = 329
3	6	1				
- 3	2	9				
	3	2				

On recommence, on abaisse la tranche suivante pour obtenir 3276. On double la racine ce qui donne $2 \times 27 = 54$ et on cherche le plus grand chiffre tel que 54_ × _ soit inférieur ou égal à 3276.

7	6	1	7	6	2	7
- 4					47	× 7 = 329
3	6	1				
- 3	2	9				
	3	2	7	6	54_ × _	

On trouve 6 et $546 \times 6 = 3276$. Le troisième chiffre de la racine est donc 6, et on retranche 3276, on obtient 0.

7	6	1	7	6	2	7	6
- 4					47	× 7 =	329
3	6	1					
- 3	2	9					
	3	2	7	6	546	× 6 =	3276
	- 3	2	7	6			
				0			

Si le reste à la fin n'est pas nul, on a obtenu la partie entière de la racine carrée de l'entier initial. Dans un premier temps nous nous arrêterons là, et nous nous contenterons de la partie entière de la racine carrée.

1. Appliquer l'algorithme pour calculer la racine de 45369. On fera apparaître sur la copie une potence sur le modèle de celle donnée dans l'exemple.
2. Écrire une fonction `decoupe` qui prend en argument un entier n et renvoie la liste de ses tranches de deux chiffres. Par exemple `decoupe(76176)` doit renvoyer `[7, 61, 76]`. Cette fonction devra être récursive.
3. Écrire une fonction `chiffre_suivant_racine` qui prend en argument `r` et `reste` deux entiers et qui renvoie le plus grand chiffre m entre 0 et 9 tels que $(10r+m) \times m$ est plus petit que `reste`.
   Par exemple, `chiffre_suivant_racine(4,361)` doit renvoyer `7`.

4. Écrire une fonction `racine_potence` qui prend en argument un entier `n` et renvoie la partie entière de sa racine carrée calculée à l'aide de la méthode de la potence.
5. On souhaite maintenant calculer une approximation de la racine carrée de `N` avec `p` chiffres exacts derrière la virgule. En remarquant que $\sqrt{a} = 10^{-p}\sqrt{10^{2p}a}$, écrire une fonction `racine_potence_precision` qui prend en argument deux entiers naturels `N` et `p` et renvoie une approximation de la racine carrée de `N` avec `p` chiffres exacts derrière la virgule. Cette fonction devra faire appel à la fonction `racine_potence`.

C- La méthode « du goutte à goutte ». Cette méthode a été mise au point en 1865 par le scientifique allemand August Joseph Toepler (1836-1912). Cette méthode repose sur le fait que la somme des n premiers entiers impairs est égale à n^2. Appliquons la méthode pour calculer la racine de 76176. On commence comme dans la méthode précédente par découper le nombre en tranches de deux chiffres à partir de la droite, ici 7|61|76. On considère la tranche la plus à gauche, et on lui retranche le plus de nombres impairs successifs possibles : $7-(1+3)=3$. On a retranché 2 nombres impairs, le premier chiffre de la racine sera donc 2.
On reporte le reste qui vaut 3 et on abaisse ensuite la tranche suivante pour obtenir 361. Le dernier impair utilisé est $i=3$: on calcule $(i+1)\times 10+1=41$ et on retranche de 361 tous les entiers impairs possibles à partir de 41 : $361-(41+43+45+47+49+51+53)=32$. On a soustrait 7 nombres impairs, le deuxième chiffre de la racine est donc 7.
Ensuite on reporte à nouveau le reste 32 et on lui accole la dernière tranche ce qui donne 3276. Le dernier nombre impair retranché étant $i=53$ on calcule $(i+1)\times 10+1=541$ et on retranche le maximum de nombres impairs à 3276 à partir de 541 : $3276-(541+543+545+547+549+551)=0$. On a retranché alors 6 nombres impairs et le dernier chiffre de la racine carrée est 6 et l'algorithme s'arrête. On a trouvé que la racine carrée de 76176 est 276.
On rassemble les étapes de l'algorithme dans le tableau ci dessous :

7|61|76			premier impair	racine
			1	
7	$-(1+3)$	$=\boxed{3}$	$(3+1)\times 10+1=41$	2
$\boxed{3}$61	$-(41+43+45+47+49+51+53)$	$=32$	$(53+1)\times 10+1=541$	27
3276	$-(541+543+545+547+549+551)$	$=0$		276

Comme dans le cas précédent, si notre nombre n'est pas un carré parfait, le reste ne sera pas nul et on obtiendra la partie entière de la racine carrée.

1. Appliquer la méthode « du goutte à goutte » pour calculer la racine carrée de 107584. On présentera le déroulé de l'algorithme dans un tableau comme dans l'exemple précédent.
2. Écrire une fonction `nombre_d_impairs` récursive prenant en argument un entier `n` et `d` un nombre impair et qui renvoie le nombre maximal de nombres impairs que l'on peut retrancher à `n` à partir de `d` ainsi que le reste. Par exemple `nombre_d_impairs(361,41)` doit renvoyer `7,32`.
3. Écrire une fonction `racine_goutte_a_goutte` qui prend en argument un entier `n` et renvoie la partie entière de sa racine carrée par la méthode « du goutte à goutte ». On pourra utiliser la fonction `decoupe` écrite à la partie précédente.

D- Et dans ma calculatrice ?
Laquelle de ces méthodes est plus probablement utilisée dans une calculatrice ? Justifier.

2 Vote par approbation

Le vote par approbation est un système de vote recommandé par de nombreuses associations mathématiques et utilisé par des outils comme Doodle. Chaque votant choisit un sous-ensemble des candidats, et le candidat choisi le plus remporte l'élection. On représente en Python le bulletin d'un votant par une liste **sans doublon**, et un vote par une liste de tels bulletins.

1. Écrire une fonction `a_un_doublon` prenant en argument une liste, et testant (en renvoyant `True` ou `False`) si elle contient un élément apparaissant au moins deux fois.
2. En déduire une fonction `vote_valide` prenant en argument une liste de listes, et testant si aucune de ces listes ne contient de doublon. Ainsi, `vote_valide` doit renvoyer :
   `True` sur `['A', 'C', 'D']`, `['A', 'B']`, `['A', 'D', 'C']`
   `False` sur `['A', 'C', 'D', 'A']`, `['A', 'B']`, `['A', 'D', 'C']`
3. Écrire une fonction `vainqueur` prenant en argument une telle liste de bulletins supposée valide, et renvoyant le vainqueur de l'élection. On utilisera un dictionnaire.
4. On s'intéresse maintenant à la question de déterminer le bulletin d'un votant en fonction de sa liste de préférences. Une stratégie simple et efficace est la suivante : Soient V et C les deux candidats favoris pour gagner l'élection, V étant le vainqueur le plus probable, il faut alors voter pour tout candidat préféré à V, et voter pour V s'il est préféré à C.
   Écrire une fonction `strategie` prenant en argument une liste de préférences (dans l'ordre décroissant), un vainqueur estimé V et un challenger estimé C, et renvoyant le bulletin décrit ci-dessus.
   Ainsi, pour une liste `Lprefs =['Alice', 'Charlie', 'Dave', 'Bob']` représentant les préférences Alice > Charlie > Dave > Bob, `strategie` doit renvoyer :
   `['Alice']` sur `Lprefs, 'Alice', 'Bob'`
   `['Alice', 'Charlie']` sur `Lprefs, 'Charlie', 'Dave'`
   `['Alice', 'Charlie', 'Dave']` sur `Lprefs, 'Bob', 'Charlie'`
5. Le vainqueur estimé et le challenger estimé peuvent provenir de sondages, qu'on peut représenter eux-mêmes comme des votes par approbation. Écrire une fonction `limite` prenant en argument la liste des listes de préférences de chaque votant, initialisant les bulletins avec uniquement le candidat préféré de chaque votant, calculant le vainqueur et son challengeur sur cette élection, modifiant les bulletins des votants en fonction de la stratégie ci-dessus, procédant à une nouvelle élection, et ainsi de suite jusqu'à ce que les bulletins des votants restent inchangés. La fonction renverra le vainqueur de l'élection lorsque cette situation d'équilibre est atteinte.