Introduction à la récursivité

Td - S1-2-1 - Récursivité

Germain Gondor

Sommaire

Sommaire

Objectif : Écrire les problèmes suivants sous formes itératives et récursive.

Objectif : Écrire les problèmes suivants sous formes itératives et récursive.

- Résoudre $f(x) = 0$ avec l'algorithme de Newton

Objectif : Écrire les problèmes suivants sous formes itératives et récursive.

- Résoudre $f(x) = 0$ avec l'algorithme de Newton
- Résoudre $f(x) = 0$ avec une méthode par dichotomie

Objectif : Écrire les problèmes suivants sous formes itératives et récursive.

- Résoudre $f(x) = 0$ avec l'algorithme de Newton
- Résoudre $f(x) = 0$ avec une méthode par dichotomie
- Palindrome

Objectif : Écrire les problèmes suivants sous formes itératives et récursive.

- Résoudre $f(x) = 0$ avec l'algorithme de Newton
- Résoudre $f(x) = 0$ avec une méthode par dichotomie
- Palindrome
- Suite de Syracuse

Sommaire

Q - 1 : *Écrire une fonction* `newton(f, fp, x0, eps)` *qui prend en arguments une fonction* `f`*, sa dérivée* `fp` *et deux flottants,* `x0` *et* `eps` *et qui renvoie une solution approchée de l'équation $f(x) = 0$ obtenue par la méthode de Newton en partant de x_0 avec un critère d'arrêt sur les images lié à* `eps`*.*

Rappel $x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)}$

Sommaire

Q - 1 : *Écrire une fonction* `dicho(f, bg, bd, eps)` *qui prend en arguments une fonction* `f` *et trois flottants* `a`, `b` *et* `eps` *et qui renvoie une solution approchée de l'équation $f(x) = 0$ sur l'intervalle $[a, b]$ en utilisant une méthode par dichotomie avec un critère d'arrêt sur les antécédents lié à* `eps`.

Sommaire

Q - 1 : *Écrire une fonction* `palindrome(mot)` *qui prend en argument une chaine de caractères* `mot` *et qui renvoie* `True` *si* `mot` *est un palindrome et* `False` *sinon.*

Sommaire

La suite de Syracuse d'un nombre entier N est définie par récurrence de la manière suivante :

$$u_0 = N \text{ et pour tout entier } n > 0 : u_{n+1} = \begin{cases} \dfrac{u_n}{2} & \text{si } u_n \text{ est pair} \\ 3.u_n + 1 & \text{si } u_n \text{ est impair} \end{cases}$$

On appelle vol de la suite de Syracuse l'ensemble des valeurs prises par la suite jusqu'à tomber sur l'entier naturel 1.

Q - 1 : *Écrire une fonction* `vol_syracuse(ui, L)` *qui prend en argument un entier* `ui` *et* `L` *une liste d'entiers et qui renvoie le vol de la suite de Syracuse.*