

TP 5 SDA Récursivité

Suites:

- faire une fonction récursive de calcul de $U_n = 2 * U_{n-1} - 1$ avec $U_0 = 5$
- afficher toutes les valeurs de U_n pour n de 0 à n (dans la fonction)
 - $U[0] = 5$
 - $U[1] = 9$
 - $U[2] = 17$
 - $U[3] = 33$
 - $U[4] = 65$
 - $U[5] = 129$
 - $U[6] = 257$
 - $U[7] = 513$
- Afficher la suite de Syracuse
 - U_0 = paramètre
 - $U_{n+1} = U_n / 2$ si U_n pair
 - $U_{n+1} = 3 * U_n + 1$ si U_n impair
 - arrêt on finit toujours par boucler sur 1 4 2 1 4 2 1 4 2 1.....
 -

Chiffres

- Afficher (un à un) les chiffres d'un nombre passé en paramètre (int)
 - 123456789 -> 1 / 2 / 3 / 4 / 5 / 6 / 7 / 8 / 9
 - Essayer en itératif
 - Puis en récursif (plus simple ?)
 - bonus: comment faire pour dépasser 10 chiffres (et pourquoi est-ce un problème ?)

Palindrome

- Proposer une (ou deux) fonction(s) récursive(s) qui vérifie qu'une chaîne est un palindrome. On pourra définir une fonction intermédiaire qui calcule des paramètres supplémentaires à passer à la fonctions récursive.

Fibonacci

- implanter fibonacci en récursifs en affichant la pile des appels (utiliser printf)
- Déduire de l'affichage de la pile que c'est inefficace (justifier)
- écrire une version itérative
- comparer le nombres d'additions

Les tours de Hanoi

On a trois poteaux sur lesquels on peut enficher des disque de différentes tailles.

Un disque ne peut être posé que sur un disque plus petit.

Comment optimalement déplacer n disques depuis le premier poteau vers le troisième.

Combien de temps pour déplacer 64 disques ? (combien de mouvements)

- implanter l'affichage des déplacements des Tours de Hanoi
- on affichera la pile comme pour Fibonacci