

Evaluation (1h) - La fonction puissance

On souhaite implanter une fonction de calcul de la puissance $(x, n) \mapsto x^n$ de plusieurs manières différentes :

Q1 (3 pts) : Programmer une fonction `int puissance_it(int x, int n)` qui calcule itérativement (en utilisant une boucle `for`) x^n à partir de x et de n .

Q2 (3 pts) : Programmer une fonction `int puissance_rec(int x, int n)` qui calcule récursivement x^n à partir de x et de n , en utilisant les règles suivantes :

$$\begin{cases} x^0 = 1 \\ x^{n+1} = x \times x^n \end{cases}$$

Q3 (3 pts) : Programmer une fonction `int puissance_dicho(int x, int n)` pour calculer par dichotomie la fonction puissance à partir de x et de n , en utilisant les règles suivantes :

$$\begin{cases} x^0 = 1 \\ x^{2n} = (x^2)^n \\ x^{2n+1} = x \times x^{2n} \end{cases}$$

Q4 (2 pts) : Programmer une fonction `int puissance_dicho2(int x, int n)` pour calculer par dichotomie la fonction puissance à partir de x et de n , en utilisant les règles suivantes :

$$\begin{cases} x^0 = 1 \\ x^{2n} = (x^n)^2 \\ x^{2n+1} = x \times x^{2n} \end{cases}$$

Q5 (3 pts) : Programmer une fonction `int *tableau_puissances(int n)` qui crée un tableau (de taille n) et le remplit avec les n premières puissances de 2. On évitera de recalculer toutes les puissances de 2 à chaque fois.

1	2	4	8	16	...	2^{n-1}
---	---	---	---	----	-----	-----------

Q6 (3 pts) : Programmer une fonction `int somme_puiss(int n)` qui utilise ce tableau pour calculer la somme des $(n + 1)$ premières puissances de 2, c'est-à-dire $\sum_{i=0}^n 2^i$. Il est interdit d'utiliser les formules mathématiques permettant de simplifier le Σ . On s'assurera que l'appel `somme_puiss(2)` produit bien 7 et que l'appel `somme_puiss(1)` produit bien 3.

Q7 (3 pts) : Programmer une fonction `int somme2_puiss(int n)` qui fait le calcul de la somme des $(n + 1)$ premières puissances de 2, c'est-à-dire $\sum_{i=0}^n 2^i$ le plus efficacement possible, et sans utiliser de tableaux pour stocker les puissances de 2. Rappel : il est interdit d'utiliser les formules mathématiques permettant de simplifier le Σ . On s'assurera que l'appel `somme2_puiss(2)` produit bien 7 et que l'appel `somme2_puiss(1)` produit bien 3.