

DS Informatique – Partie écrite. Durée : 1 heure

Lorsqu'on écrit un code Python : faire attention à ce que les indentations soient visibles sur la copie ; commenter le code de façon à expliquer les grandes étapes de l'algorithme en ajoutant un commentaire en fin de ligne de code après le symbole `#`.

## I Question de TP – Comparaison entre deux méthodes d'exponentiation

On propose deux programmes différents pour l'exponentiation d'un réel positif  $k$  par un entier strictement positif  $n$  (c'est-à-dire le calcul de  $k^n$ ). On suppose que  $k$  est affecté d'une valeur réelle positive.

---

```

1 p = 1
2 c = n
3 while c > 0:
4     p = p * k
5     c = c - 1

```

---

Programme 1: Première méthode d'exponentiation.

---

```

1 p = 1
2 c = n
3 while c > 0:
4     if c%2 == 1:
5         p = p * k
6     k = k**2
7     c = c//2

```

---

Programme 2: Deuxième méthode d'exponentiation.

1. Pour le programme 1, identifier et prouver le variant et l'invariant de boucle.
2. On prend  $n = 13$ . Pour chacun des deux programmes 1 et 2, déterminer et noter les valeurs ou expressions successives de  $p$ ,  $c$  et  $k$  ainsi que le nombre d'itérations de la boucle Tant que.
3. Compter le nombre d'opérations élémentaires dans chacune des boucles. En déduire lequel des deux programmes réalise une « exponentiation rapide ».
4. On peut montrer que la *complexité temps* du programme d'exponentiation rapide est  $O(\log(n))$ . Quelle est la *complexité temps* de l'autre programme ?

## II Question de cours – Algorithme d'Euclide

Soient  $a \in \mathbb{N}$  et  $b \in \mathbb{N}^*$ . Donner le code en python d'une fonction qui réalise l'algorithme d'Euclide déterminant le plus grand diviseur commun de  $a$  et  $b$ .

## III Exercice – Expression de durées

On suppose déjà connue du programme deux listes L1 et L2 auxquelles sont affectées des durées  $d_1$  et  $d_2$  en jours, heures, minutes et secondes sous la forme de quatre entiers `[jours,heures,minutes,secondes]`. Donner un code python permettant d'exprimer la durée totale  $d = d_1 + d_2$  en jours, heures, minutes et secondes de façon unique. Par exemple : pour L1=[1,23,45,57] et L2=[1,4,17,28], le résultat doit être la liste `[3,4,3,25]`.

Piste suggérée : d'abord convertir les durées en secondes puis convertir la somme des durées en jours, heures, minutes et secondes.