– I –

1. Écrire deux fonctions, un itérative et une récursive, calculant la somme des `N` premiers cubes d'entiers naturels.
2. Écrire une fonction récursive d'entête `somPart(u,n,N)` prenant comme arguments une liste `u` et deux entiers `n` et `N` et renvoyant la valeur de la somme partielle
$$\sum_{k=n}^{N} u[k] \,.$$
3. Écrire une fonction récursive d'entête `somPartF(f,n,N)` prenant comme arguments une fonction python `f` et deux entiers `n` et `N` et renvoyant la valeur de la somme partielle
$$\sum_{k=n}^{N} f(k) \,.$$
4. Utiliser les algorithmes précédents pour déterminer la valeur des sommes partielles suivantes :
$$\sum_{k=0}^{42} \frac{k^3+2}{k+1}, \qquad \sum_{k=1}^{80} \frac{1}{k^2} \qquad \text{et} \qquad \sum_{k=18}^{145} k(k+1)(k+3) \,.$$

– II –

1. Écrire une fonction d'entête `expNaive(x,N)` calculant de façon récursive le nombre x^N. Quelle est sa complexité ?
2. Écrire une fonction d'entête `expRap(x,N)` calculant le nombre x^N par la méthode de l'*exponentiation rapide*, à savoir :
   — si $N = 0$, renvoyer 1 ;
   — sinon, si N est pair renvoyer $(x^2)^{N/2}$ (calculé récursivement) ;
   — sinon, si N est impair renvoyer $x * (x^2)^{(N-1)/2}$ (calculé récursivement).
   Quelle est la complexité de cette fonction ?