TP n^{o}3 : Fonctionnelles

Q1 Somme et série

(a) Définir une fonctionnelle `somme_termes` qui calcule la somme des $n+1$ premiers termes d'une suite (u_n) de nombres réels.
Utiliser `somme_termes` pour faire la somme des 100 premiers *entiers*.
Vérifiez que le résultat est bien 5050.

Utiliser `somme_termes` pour faire la somme des nombres *impairs* $\leqslant 10$
Vérifiez que le résultat est bien 25.

(b) Utiliser `somme_termes` pour calculer la somme de la suite
$1 - 2 + 3 - 4 \ldots + 99$. Vérifiez que le résultat est bien 50.

(c) Définir la fonction `inv100` telle que
$inv100(x) = (1-x)^0 + (1-x)^1 + (1-x)^2 + \ldots + (1-x)^{100}$
Vérifiez que le résultat est bien 2.0 pour $x = 0.5$

Q2 Nombres parfaits

(a) Définir une fonctionnelle `somme_filtre` qui, étant donné un nombre n et une fonction f renvoyant un booléen , calcule la somme des entiers p tels que $p \leqslant n$ et $f(p) = True$
Vérifiez que le résultat est bien 30 pour les nombres pairs $\leqslant 10$

(b) Un nombre parfait est un entier
$\neq 0$
$=$ à la moitié de la somme de ses diviseurs
Exemple : 6 est un nombre parfait car
ses diviseurs sont $1, 2, 3$ *et* 6
la somme de ses diviseurs est $1 + 2 + 3 + 6 = 12$
et la moitié de la somme de ses diviseurs est $\frac{12}{2} = 6$
Utiliser la fonction `somme_filtre` pour définir une fonction `est_parfait` qui teste si un nombre est parfait.

Q3 Application itérée

(a) Définir une fonctionnelle `applyn` qui, étant donné un nombre n, une fonction f et une valeur x, calcule le résultat de f appliquée n fois sur x, c'est à dire $f^n(x) = \underbrace{f(\ldots(f(f}_{n\ fois}(x))))$

Vérifiez votre fonction, par exemple que l'application de 5 fois la fonction $(x+1)$ avec $x = 10$ donne bien 15

(b) Utiliser la fonction `apply` pour définir la fonction `power` qui calcule x^n

Q4 Matrices fonctionnelles (*ou fonctions de matrices*)

On choisit de représenter une matrice M de dimensions $L \times C$ par une fonction $f(i, j)$. Cette fonction donne en résultat un couple de la forme :

- $(vrai, M_{ij})$ si i et j sont valides
  c'est à dire si $1 \leqslant i$ et $i \leqslant L$ et $1 \leqslant j$ et $j \leqslant C$
- $(faux, 0)$ sinon

Exemple : la matrice ci-dessous

$$\begin{pmatrix} 3 & 4 & 5 & 6 & 7 \\ 5 & 6 & 7 & 8 & 9 \\ 7 & 8 & 9 & 10 & 11 \\ 9 & 10 & 11 & 12 & 13 \\ 11 & 12 & 13 & 14 & 15 \\ 13 & 14 & 15 & 16 & 17 \end{pmatrix}$$

est représentée par la fonction qui à tout couple (i, j) associe

- $(vrai, 2 \times i + j)$ si $1 \leqslant i$ et $i \leqslant 6$ et $1 \leqslant j$ et $j \leqslant 5$
- $(faux, 0)$ sinon

(a) Écrivez la fonction `exemple` représentant la matrice de l'exemple

(b) Écrivez la fonction `identite_4_4` représentant une matrice 4×4 avec des 1 sur sa diagonale et des 0 partout ailleurs.

(c) Définissez la factorielle `dims(M)` qui renvoie un couple (L, C) correspondant aux dimensions d'une matrice représentée par la fonction `M`.
Il faut donc ici chercher les plus petits indices de ligne et colonne pour lesquels la fonction `M` renvoie un résultat de la forme $(vrai, \ldots)$

(d) Définissez une fonction `add_mat(mat_A,mat_B)` qui retourne une fonction correspondant à la somme des matrices A et B qui elles, sont représentées par leur fonctions `mat_A` et `mat_B`), en sachant que :
   - on ne peut faire la somme de 2 matrices que si elles ont les mêmes dimensions
   - que par définition $A + B = (a_{ij} + b_{ij})$

(e) Testez `add_mat(exemple, identite_4_4)`. Pourquoi ce résultat ?

(f) Faites une autre fonction `identite_6_5` pour une matrice de 6×5 et vérifiez `add_mat(exemple, identite_6_5)`