Exercice 2

Cet exercice porte sur la programmation en général et la récursivité en particulier.

On considère un tableau de nombres de n lignes et p colonnes.

Les lignes sont numérotées de 0 à $n-1$ et les colonnes sont numérotées de 0 à $p-1$. La case en haut à gauche est repérée par $(0,0)$ et la case en bas à droite par $(n-1,p-1)$.

On appelle *chemin* une succession de cases allant de la case $(0,0)$ à la case $(n-1,p-1)$, en n'autorisant que des déplacements case par case : soit vers la droite, soit vers le bas.

On appelle *somme* d'un chemin la somme des entiers situés sur ce chemin.

Par exemple, pour le tableau `T` suivant :

4	**1**	**1**	3
2	0	**2**	1
3	1	**5**	**1**

— Un chemin est $(0,0)$, $(0,1)$, $(0,2)$, $(1,2)$, $(2,2)$, $(2,3)$ (en gras sur le tableau) ;

— La somme du chemin précédent est 14.

— $(0,0)$, $(0,2)$, $(2,2)$, $(2,3)$ n'est pas un chemin.

L'objectif de cet exercice est de déterminer la somme maximale pour tous les chemins possibles allant de la case $(0,0)$ à la case $(n-1,p-1)$.

Question 1 On considère tous les chemins allant de la case $(0,0)$ à la case $(2,3)$ du tableau `T` donné en exemple.

1. Un tel chemin comprend nécessairement 3 déplacements vers la droite. Combien de déplacements vers le bas comprend-il ?
2. La longueur d'un chemin est égal au nombre de cases de ce chemin. Justifier que tous les chemins allant de $(0,0)$ à $(2,3)$ ont une longueur égale à 6.

Question 2 En listant tous les chemins possibles allant de $(0,0)$ à $(2,3)$ du tableau `T`, déterminer un chemin qui permet d'obtenir la somme maximale et la valeur de cette somme.

Question 3 On veut créer le tableau `T'` où chaque élément `T'[i][j]` est la somme maximale pour tous les chemins possibles allant de $(0,0)$ à (i,j).

1. Compléter et recopier sur votre copie le tableau `T'` donné ci-dessous associé au tableau

`T` =

4	1	1	3
2	0	2	1
3	1	5	1

.

`T'` =

4	5	6	?
6	?	8	10
9	10	?	16

2. Justifier que si j est différent de 0, alors : T' [0] [j] = T [0] [j] + T' [0] [j-1]

Question 4 Justifier que si i et j sont différents de 0, alors : T' [i] [j] = T [i] [j] + max(T' [i-1] [j], T' [i] [j-1]).

Question 5 On veut créer la fonction récursive `somme_max` ayant pour paramètres un tableau T, un entier i et un entier j. Cette fonction renvoie la somme maximale pour tous les chemins possibles allant de la case (0, 0) à la case (i, j).

1. Quel est le cas de base, à savoir le cas qui est traité directement sans faire appel à la fonction `somme_max` ? Que renvoie-t-on dans ce cas ?
2. À l'aide de la question précédente, écrire en Python la fonction récursive `somme_max` .
3. Quel appel de fonction doit-on faire pour résoudre le problème initial ?