Option Informatique en Sup MPSI
TP : Sudoku

▶ *Sudoku* est un casse-tête qui nous vient, paraît-il, de l'Orient mystérieux… L'objectif est de remplir avec les chiffres 1 à 9 les cases vides d'un tableau comme celui de gauche, tout en respectant la règle suivante : dans chaque ligne, chaque colonne et chaque «bloc», les chiffres 1 à 9 doivent apparaître une fois (et, donc, une seule).

1	3					8		
			5	3			2	
				8				6
		6				5		
	7	3				6	8	
		9				4		
4				7				
	6			5	2			
		7					4	1

0	**1**	**2**
3	**4**	**5**
6	**7**	**8**

1	3	5	2	4	6	8	9	7
6	9	8	5	3	7	1	2	4
7	2	4	1	8	9	3	5	6
8	4	6	7	2	3	5	1	9
5	7	3	9	1	4	6	8	2
2	1	9	8	6	5	4	7	3
4	8	2	3	7	1	9	6	5
9	6	1	4	5	2	7	3	8
3	5	7	6	9	8	2	4	1

Le tableau de droite donne une solution de ce casse-tête ; on peut montrer qu'il n'y en a pas d'autre.

▶ Désormais, nous appelerons *sudoku d'ordre 3* (ou tout simplement *sudoku*) une grille comme celle de droite. Notez qu'un sudoku d'ordre n contient n^4 cases.

▶ Les lignes sont numérotées de 0 à 8 (en allant du haut vers le bas) et les colonnes de 0 à 8 (en allant de la gauche vers la droite), conformément à la tradition Caml. Les neuf blocs sont numérotés de 0 à 8, selon le petit tableau central.

Question 1 Montrez que la case située ligne 1, colonne 0 doit contenir un 6.

Un peu de programmation

▶ Un sudoku sera représenté par un objet `m` de type `int vect vect`. Ainsi, un fragment de programme construisant le sudoku de notre exemple pourrait comporter (entr'autres) les lignes suivantes :

```
let m = make_matrix 9 9 0 ;;
m.(0).(0) <- 1 ;;
...
m.(5).(2) <- 9 ;;
...
m.(8).(9) <- 1 ;;
```

Question 2 • Rédigez en Caml une fonction de signature :

```
flat : 'a list list -> 'a list
```

spécifiée comme suit : `flat l` «aplatit» la liste de listes ℓ ; ainsi, `flat [[2;7];[9;1;3];[3];[];[5;2]];;` rendra la liste `[2;7;9;1;3;3;5;2]`. Objectif : trois lignes.

▶ Pour les questions suivantes, on ne vous demande pas de tester la validité des arguments : g, d, h et b seront censés appartenir à l'intervalle $[\![0, 8]\!]$ et vérifier $g \leqslant d$ et $h \leqslant b$; m se devra d'être une brave matrice carrée d'ordre 9.

Question 3 • Rédigez en Caml une fonction de signature :

```
list_of_bande : int -> int -> int -> 'a vect vect -> 'a list
```

spécifiée comme suit : `list_of_bande h g d m` construit la liste (présentée dans un ordre quelconque) des $m_{h,j}$ où j décrit $[\![g, d]\!]$. Objectif : deux lignes ; vous pouvez utiliser `intervalle`, à condition de rappeler sa définition.

Question 4 • Rédigez en Caml une fonction de signature :

```
list_of_zone : int -> int -> int -> int -> 'a vect vect -> 'a list
```

spécifiée comme suit : `list_of_zone h b g d m` construit la liste (présentée dans un ordre quelconque) des $m_{i,j}$ où (i,j) décrit $[\![h,b]\!] \times [\![g,d]\!]$. Objectif : deux lignes.

Question 5 • Rédigez en Caml une fonction de signature :

```
list_of_bloc : int -> 'a vect vect -> 'a list
```

spécifiée comme suit : `list_of_bloc k m` construit la liste (présentée dans un ordre quelconque) des $m_{i,j}$ où (i,j) décrit le bloc numéro k. Objectif : trois lignes.

Question 6 • Rédigez en Caml une fonction de signature :

```
bonne_liste : int list -> bool
```

spécifiée comme suit : `bonne_liste l` indique si la liste ℓ contient, à l'ordre près, les entiers de 1 à 9. Objectif : deux lignes, si possible !

Question 7 • Rédigez en Caml une fonction de signature :

```
bon_sudoku : int vect vect -> bool
```

spécifiée comme suit : `bon_sudoku m` indique si le tableau m contient effectivement un sudoku. Objectif : sept lignes.

FIN