

Consignes

Le fichier dont il est question dans le sujet se trouve dans le répertoire `c:\AP2\`. Vous devez le compléter pour répondre aux questions posées. Le fichier complété ainsi que les différents fichiers sources doivent être rendus en version électronique par email à `leon@sic.univ-poitiers.fr`.

Le titre du mail doit être au format *AP2 - Nom1 Prénom1, Nom2 Prénom2*.

Vous écrirez au début de vos fichiers sources ADA les mentions suivantes :

```
-----
-- AP2 - GROUPE PR - Mardi 8 Décembre 2007 - 1?h00/1?h00
-- NOM1, prénom1, No carte étudiant
-- NOM2, prénom2, No carte étudiant
-----
```

Sujet

Le sudoku... un beau jeu en soit, mais qui peut prendre beaucoup de temps à résoudre. C'est justement ce dont manque votre prof d'informatique! Il s'adonne néanmoins à ce jeu de temps à autre avec son ami Michael¹ qui lui vente la supériorité des mathématiciens sur les informaticiens (guerre aussi violente, heu je veux dire bon enfant qu'entre une mathématicien et un physicien...). Or Michael m'a posé une grille assez démoniaque : "une demi-heure tout au plus pour Nous" m'a-t-il balancé à la figure avec un sourire en coin. Ces temps-ci n'ayant, ni la patience, ni le temps de résoudre son défi, je vous propose, à vous mes meilleurs éléments, de m'aider dans cette lourde tâche grâce à un petit programme informatique dont je vous donne la structure que j'ai tapée entre deux présentations lors d'une conférence (autant joindre l'utile, à...). Voici la

grille du diable² :

0	0	0	9	0	0	5	4	7
0	0	3	1	7	0	0	0	0
0	0	0	0	0	0	1	0	0
0	3	0	0	0	0	0	5	2
0	0	4	0	6	0	7	0	0
1	5	0	0	0	0	0	9	0
0	0	1	0	0	0	0	0	0
0	0	0	0	2	7	6	0	0
8	6	5	0	0	9	0	0	0

où les "0" représentent les valeurs à trouver.

Pour rappel, "la grille de jeu est un carré de neuf cases de côté, subdivisé en autant de carrés identiques, appelés régions (voir figure). La règle du jeu est simple : chaque ligne, colonne et région ne doit contenir qu'une seule fois tous les chiffres de un à neuf. Formulé autrement, chacun de ces ensembles doit contenir tous les chiffres de un à neuf." (Wikipédia)

Pour résoudre ce problème, je vous propose de le décomposer en sous problèmes plus simples à résoudre.

Question 1 : Écrire une procédure `Put` qui affiche une grille `G` à l'écran (faites un affichage simple!).

Question 2 : Écrire une fonction `Intersection` qui à partir de deux piles triées `P1`, `P2` (la valeur max est au sommet de la pile) retourne une pile `P` contenant l'intersection des valeurs de `P1` et `P2` (attention : la pile `P` doit être triée dans le même sens à la sortie de la fonction). Exemple : si `P1 = [1, 2, 5, 7]` et `P2 = [3, 4, 5, 7, 8]` alors `P = [5, 7]`.

Question 3 : Écrire une procédure `RemplirPile` qui à partir d'un tableau `T` : `TblSol` de booléens empile sur une pile `P` les valeurs des indices où `T(i)` est vrai. Exemple : si `T = [True, False, False, True]` alors `P = [1, 4]`.

¹Michael : nom hébraïque qui signifie "Qui est comme Dieu" / "Qui est comme un ange" suivant la traduction

²diabale : du grec diabolos qui veut dire séparer

Question 4 : Écrire une fonction `TrouverSolCol` qui à partir d'une grille `G` et d'une colonne `X`, retourne une Pile `P` contenant les chiffres non utilisés dans la colonne `X`.

Question 5 : Écrire une fonction `TrouverSolLig` qui à partir d'une grille `G` et d'une ligne `Y`, retourne une Pile `P` contenant les chiffres non utilisés dans la ligne `Y`.

Question 6 : Écrire une fonction `TrouverSolReg` qui à partir d'une grille `G`, d'une colonne `X` et d'une ligne `Y`, retourne une Pile `P` contenant les chiffres non utilisés dans la région contenant la case `(X, Y)`. (Attention c'est plus difficile)

Question 7 : Dédurre des questions 2, 4, 5 et 6 une fonction qui retourne une pile `P` contenant tous les chiffres potentiels pour une case `(X, Y)` donnée.

Question 8 : Écrire une procédure qui à partir d'une grille `G`, renvoie une case `(X, Y)` vide. (i.e. qui contient la valeur 0). Si la grille ne contient aucune case vide, renvoyez `(0, 0)`.

Question 9 : Mélangeons le tout ensemble... écrire la procédure `Solve` qui à partir d'une grille `G` renvoie une grille `Gp` résolue. (et montrons à notre mathématicien qu'en deux heures vous avez généralisé la recherche d'une solution d'une grille de sudoku et qu'il vous suffit maintenant de quelques secondes pour résoudre n'importe quelle grille contre 30 mins pour lui le temps qu'il remette en route tous ses théorèmes).

Pour cela, vous allez commencer par rechercher la première case vide `(X, Y)` de `G` (que se passe-t-il s'il n'y a plus de case vide?). Ensuite, vous allez essayer de résoudre une nouvelle grille `nG`, où la case vide est remplacée par chacune des valeurs potentielles de la case `(X, Y)`. (Inspirez... (ou soyez-le), ne sentez-vous pas comme une odeur de récursif?)

Question 10 : Quelle est la hauteur maximale de l'arbre d'appels de procédures?

Références :

- <http://hal.inria.fr/docs/00/08/58/09/PDF/47.pdf> : article "Évaluer la difficulté d'une grille de sudoku à l'aide d'un modèle contraintes" ;
- <http://fr.wikipedia.org/wiki/Sudoku>.

N'oubliez pas... je compte sur vous!

(Attention : ceci n'est qu'une fiction aucun lieu ou personnage n'est réel. Aucun informaticien, physicien, mathématicien n'a été tué ou blessé durant la rédaction de ce sujet.)