

## TP2 AGP : Su Doku

Durée encadrée prévue : 4 heures

Pas de compte rendu, faire valider la solution par l'enseignant

### 1 Programmation : résolution de Su Doku

Une grille de Su Doku est une grille de taille  $9 \times 9$ , subdivisée en 9 carrés  $3 \times 3$  appelés *régions*. Les cases de cette grille peuvent recevoir des chiffres. Le jeu de Su Doku consiste à compléter une grille partiellement remplie par des chiffres en respectant la condition suivante : chaque ligne, colonne et région ne doit contenir qu'une seule fois tous les chiffres de un à neuf.

Un exemple de grille est représentée sur la gauche de la figure 1, la grille complétée est représentée sur la droite de la même figure. On peut vérifier que sur la grille de droite, chaque ligne contient tous les chiffres une et une seule fois, il en est de même pour chaque colonne et chaque région.

Dans cette partie on utilisera un tableau bidimensionnel d'entiers de taille  $9 \times 9$  pour stocker une grille de Su Doku. Des fonctions vous sont proposées pour lire et écrire cette structure dans un fichier texte : téléchargez l'archive TP2.tar sur Moodle.

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

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

FIG. 1 – Une exemple de grille de Su Doku avec sa solution à droite

## 1.1 Vérification de solution

- Écrire une fonction C qui vérifie qu'une grille complètement remplie est un Su Doku valide, la fonction renvoie 1 si le Su Doku est valide, 0 sinon. Voici le prototype de la fonction demandée :

```
int sudokuValide(int sudoku[9][9])
```

## 1.2 Résolution de Su Doku simples

Pour cette partie, nous allons supposer que nous n'avons que des grilles *simples* de Su Doku à résoudre (la définition de grille simple est donnée plus loin). On se propose d'implémenter la méthode suivante pour résoudre les Su Doku simples :

1. pour chaque chiffre *i* présent dans une case *c*, on mémorise le fait que ce chiffre *i* ne peut pas être mis dans une case présente sur la même ligne que celle de *c* ou sur même colonne que celle de *c* ou sur la même région que celle de *c*.
2. Compte-tenu de ces interdictions, on cherche une case vide pour laquelle il existe une *unique* valeur permise.
3. Si on ne trouve pas une telle case l'algorithme échoue (le Su Doku n'est pas *simple*).
4. Si on trouve une telle case : on la remplit, on mémorise les nouvelles interdictions introduites par le chiffre qu'on a rajouté et on repart en 2.

Un Su Doku sera donc dit *simple* si cette méthode permet de le résoudre, c'est à dire qu'à chaque étape il existe une case pour laquelle une seule valeur est possible. La grille de la figure 1 est un Su Doku simple : la figure 2 montre trois premières étapes possibles pour commencer le remplissage de la grille : la case de coordonnées (8,5) ne peut contenir que la valeur 1, ensuite la case de coordonnées (8,7) ne peut contenir que la valeur 5, ce qui entraîne que la case de coordonnées (7,7) ne peut contenir que la valeur 3 etc...

		4		3	6			5			4		3	6			5			4		3	6			5
		5	1				8				5	1				8				5	1				8	
1	3	7	8		4	6		2	1	3	7	8		4	6		2	1	3	7	8		4	6		2
	5		4				3	6		5		4				3	6		5		4				3	6
4	9			7	3	2		8	4	9			7	3	2		8	4	9			7	3	2		8
		6		8		9	7				6		8		9	7				6		8		9	7	
8		9	7	6	5			1	8		9	7	6	5			1	8		9	7	6	5	<b>3</b>		1
6		2	3	<b>1</b>			4		6		2	3	1		<b>5</b>	4		6		2	3	1		5	4	
			9			8		7				9			8		7				9			8		7

FIG. 2 – Trois premières cases remplies par notre méthode (notez qu'il y a d'autres choix possibles à chaque étape)

- Écrire une fonction C qui prend en entrée une grille partiellement remplie, et la remplit complètement avec cette méthode (ou échoue si le Su Doku n'est pas simple). Pour stocker les interdictions, on pourra par exemple utiliser un tableau d'entiers de dimension 3, chaque dimension étant de taille 9 :

```
int M[9][9][9]
```

La case `M[i][j][k]` vaudra 1 si le chiffre `k+1` ne peut pas être écrit dans la case de coordonnées `(i+1,j+1)` et 0 sinon (les tableaux en C commençant à 0, il faut ajouter 1 pour avoir des valeurs entre 1 et 9). Il n'y a pas obligation d'utiliser cette structure de données, vous pouvez proposer une autre structure de données pour écrire cette fonction.

### 1.3 Algorithme de résolution

Proposer un algorithme de résolution de Su Doku quelconque (pas forcément simples)