TP1 : Programmation Par Contraintes

Exercice 1

L'objectif de cette première étude est multiple :
— Réfléchir à une solution pour résoudre un problème combinatoire : `sudoku(`n`)` .
— Initiation à la PPC avec la bibliothèque et le solveur `choco`.
— Comparaison entre une solution impérative et une solution déclarative.

Vous trouverez dans votre dépôt local :
— `"fr.univ_montpellier.fsd.sudoku.ppc.Sudoku.java"` : une modélisation PPC du problème `sudoku(`n`)` .
— `"fr.univ_montpellier.fsd.sudoku.ppc.HardSudoku.java"` : une instance sudoku à modéliser.
— `"fr.univ_montpellier.fsd.sudoku.ppc.GTSudoku.java"` : une variante du problème à modéliser.
— `"fr.univ_montpellier.fsd.sudoku.imp.Sudoku.java"` : une solution Java à implémenter du problème.
— `fr.univ_montpellier.fsd.sudoku.App.java"` : module pour comparer les solutions.

Question 1 • Proposez une solution impérative et son implémentation dans `"imp.Sudoku.java"` pour résoudre `sudoku(`n`)` .

Question 2 • Modélisez le problème sous la forme d'un réseau de contraintes.

Question 3 • Analysez et testez le modèle à contraintes dans `"ppc.Sudoku.java"`, corrigez les erreurs de modélisation.

Question 4 • Comparez les deux solutions dans l'environnement `App.java`, quelles sont vos conclusions ?

Question 5 • Révisez les deux solutions (Java / PPC) pour retourner l'ensemble des solutions.

Nous allons passer maintenant à la modélisation PPC avec l'idée de voir comment on peut réviser un modèle sans difficulté en PPC. Voici une des plus difficiles instances du sudoku :

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

Question 6 • Révisez le modèle PPC qui est dans `HardSudoku.java` pour résoudre l'instance donnée avant.

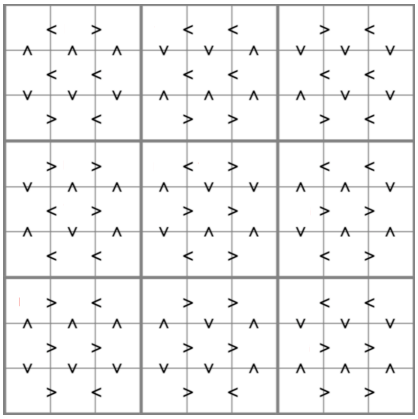
Question 7 • Ci-dessous une instance 16×16. Proposez une façon simple pour pouvoir passer du modèle `HardSoduku.java` à celui de l'instance 16×16 (`VeryHardSoduku.java`) en modifiant quelques lignes de code.

	G			F	8	9	6	4	B	D	5			3	
6	C					4	E	2	7					5	9
			D			G	7	F	E			6			
		4	3	A							6	1	B		
7			5	8	F					B	E	9			G
8				9			4	D			3				2
C	1	3				6			G				F	4	5
9	D	B			G					F			7	A	6
G	B	A			2					7			5	6	D
5	6	F				A			2				8	7	4
D				6			9	5			G				F
3			C	B	5					A	4	G			1
		9	6	G							7	2	C		
			G			B	D	C	5			F			
4	3					8	2	G	F					1	7
	8			5	9	E	A	1	3	2	D			G	

Question 8 • Retournez l'ensemble des solutions des deux instances.

Une des variantes du sudoku est le `Greater Than Sudoku`.

Question 9 • Révisez le modèle dans `GTSUdoku.java` de sorte à résoudre l'instance suivante :



Exercice 2

Dans cette étude, nous allons colorier la carte des 13 régions de France de sorte que deux régions ayant une frontière en commun soient coloriées différemment. Pour cela, nous disposons de quatre couleurs : `bleu`, `rouge`, `vert` et `jaune`.

Question 1 • Proposez une solution impérative pour résoudre le problème.

Question 2 • Modélisez le problème sous la forme d'un réseau de contraintes N.

Question 3 • Implémentez N sous choco.

Question 4 • Comparez les deux approches sur l'instance des régions.

Question 5 • Lancez les deux approches sur l'ensemble des instances du dossier `datasets` et présentez une étude comparatives avec des observations et des conclusions.