Université d'Ottawa
Faculté de génie

École de science
d'informatique
et de génie électrique

uOttawa
L'Université canadienne
Canada's university

University of Ottawa
Faculty of Engineering

School of Electrical
Engineering
and Computer Science

Assignment 2
CSI2120 Programming Paradigms

Winter 2022

Due March 30 at 11:30 pm – Accepted late with penalty till April 1

8%

Mini-Sudoku

2	1	4	3
4	3	2	1
1	2	3	4
3	4	1	2

Un mini-sudoku est un tableau de 4x4 dans lequel chaque élément est l'un de quatre nombres possibles 1, 2, 3, 4. Pour être Sudoku valide, chacune de ses rangées, de ses colonnes et de ses quadrants doivent contenir des nombres différents. Un Sudoku valide est illustré ci-dessus.

Pour cette question, vous devez écrire un programme prenant en entrée un Sudoku (une matrice 4x4) et qui vérifie si ce Sudoku est valide ou non (la fonction retourne donc un booléen). Notez bien que le Sudoku donné est déjà rempli, vous ne faites que vérifier sa validité.

Le Sudoku doit être représenté à l'aide d'une liste de 4 listes, chacune des sous-listes représentant une rangée du Sudoku.

```
sudoku([[2,1,4,3],[4,3,2,1],[1,2,3,4],[3,4,1,2]]).
sudoku([[2,1,4,3],[4,3,2,1],[1,2,3,3],[3,4,1,2]]).
```

A mini-sudoku is an array of 4x4 in which each entry is one of the four numbers 1,2,3,4. To be a valid Sudoku, each row, each column, and each of the four quadrants must contain different numbers (as shown in the figure above). Note that this Sudoku is already complete, you only have to check its validity.

We ask you to write a program that will take as input a completed Sudoku (a 4x4 matrix) and will check if this one is a valid Sudoku solution (true or false).

The Sudoku must be represented with a list of 4 lists, each list element representing one row of the matrix.

```
sudoku([[2,1,4,3],[4,3,2,1],[1,2,3,4],[3,4,1,2]]).
sudoku([[2,1,4,3],[4,3,2,1],[1,2,3,3],[3,4,1,2]]).
```

Question 1: [1 pt]

Définir le prédicat `different/1` qui est vrai lorsque tous les nombres d'une liste sont différents.

Write the predicate `different/1` that is true if all numbers in a list are different.

```
?- different([1,3,6,4,8,0]).
yes
?- different([1,3,6,4,1,8,0]).
no
```

Question 2: [1.5 pt]

Définir le prédicat `extract4Columns/2` qui extrait les 4 colonnes d'un mini-Sudoku.

Write the predicate `extract4Columns/2` that extracts the 4 columns of the 4x4 mini-Sudoku.

```
?-  sudoku(M),  extract4Columns(M,L).
L=[[2,4,1,3],[1,3,2,4],[4,2,3,1],[3,1,4,2]])
```

Question 3: [1.5 pt]

Définir le prédicat `extract4Quadrants/2` qui extrait les 4 quadrants d'un mini-Sudoku.

Write the predicate `extract4Quadrants/2` that extracts the 4 quadrants of the 4x4 Sudoku.

```
?-  sudoku(M),  extract4Quadrant(M,L).
L=[[2,1,4,3],[4,3,2,1],[1,2,3,4],[3,4,1,2]])
```

Question 4: [2 pts]

Définir le prédicat `allDifferents/1` qui vérifie si chacune des sous-listes d'une liste de listes contient des nombres différents. Utiliser le prédicat `different/1` de la question a) dans la définition de ce prédicat.

Write the prdicate `allDifferents/1` that checks if each sublist of a list of list contains different numbers. Use predicate `different/1` from question a) to define this predicate.

```
?- different([[1,3,6,4,8,0],[1,3,6,4,1,8,0]]).
no
```

Question 5: [2 pts]

Afin de vérifier la validité d'un Sudoku, vous avez simplement à vérifier si toutes les sous-listes de la représentation du Sudoku sont toutes différentes (`alldifferents`) et si toutes les sous-listes obtenues à partir de `extract4Columns` et `extract4Quadrants` sont aussi toutes différentes (`alldifferents`).
Définir le prédicat `checkSudoku/1` en vous basant sur cette stratégie.

In order to verify the validity of a Sudoku, you just have to check if each of the lists in the sudoku representation are `alldifferents` and if the lists from `extract4Columns` and `extract4Quadrants` are also `alldifferents`.
Write the predicate `checkSudoku/1` that uses this strategy.

```
?- sudoku(M),checkSudoku(M).
yes;
no.
```