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 3

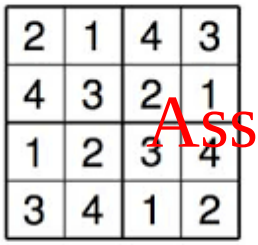
CSI2120 Programming Paradigms

Winter 2023

Due on April 10th 2023, 11 :30 PM

8% - 10 points

Mini-Sudoku



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).

Question 1:

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

```
(define sudoku1 '((2 1 4 3) (4 3 2 1) (1 2 3 4) (3 4 1 2)))
(define sudoku2 '((2 1 4 3) (4 3 2 1) (1 2 3 3) (3 4 1 2)))
```

You must solve the problem using the following functions:

a) *Write the function* `different` *that returns true if all numbers in a list are different. [2]*

```
(different '(1 3 6 4 8 0))
#t
```

```
(different '(1 3 6 4 1 8 0))
#f
```

b) *Write the function* `extract4Columns` *that extracts the 4 columns of the 4x4 Sudoku. [2]*

```
(extract4Columns sudoku1)
((2 4 1 3) (1 3 2 4) (4 2 3 1) (3 1 4 2))
```

c) *Write the function* `extract4Quadrants` *that extracts the 4 quadrants of the 4x4 Sudoku. [2]*

```
(extract4Columns sudoku1)
((2 1 4 3) (4 3 2 1) (1 2 3 4) (3 4 1 2))
```

d) *Write the function* `merge3` *that merges three lists. [2]*

```
(merge3 '(1 3 6) '(5 4) '(1 2 3))
(1 3 6 5 4 1 2 3)
```


e) *Write the function* `checkSudoku` *that verifies if a sudoku is valid by proceeding as follows:* [2]

- *merges together the list of rows of the sudoku with the list of its columns (from* `extract4Columns`*) and the lists of its quadrants (from* `extract4Quadrants`*);*
- *use* `map` *in order to call the function* `different` *on each element of the merged list;*
- *use the result of* `map` *in order to determine if the sudoku is valid and return the result (true or false).*


```
(checkSudoku sudoku1)
#t
(checkSudoku sudoku2)
#f
```