

# info803 : informatique

## TP 3 : Sudoku

Pierre Hyvernats  
Laboratoire de mathématiques de l'université de Savoie  
bâtiment Chablais, bureau 22  
téléphone : 04 79 75 94 22  
email : [Pierre.Hyvernats@univ-savoie.fr](mailto:Pierre.Hyvernats@univ-savoie.fr)  
www : <http://www.lama.univ-savoie.fr/~hyvernats/>

*Le compte-rendu de TP sera contenu en commentaire dans les sources de votre programme. Vous devrez donc me rendre un unique fichier appelé "nom.c". Votre fichier devra être le plus lisible possible (indentation intelligente de votre code, présentation lisible de vos commentaires).*

### Exercice 0 : représentation des flottants (30 minutes)

Récupérez le fichier `flottants.c` sur ma page web... Utilisez ce fichier pour explorer la représentation des nombres flottants. (Norme IEEE 754.)

Faites la question 3 de l'exercice 3 du TP 2...

### Exercice 1 : Sudoku

J'espère que tout le monde connaît le principe du Sudoku : il faut remplir un carré de 81 cases ( $9 \times 9$ ) en respectant les contraintes suivantes

- tous les chiffres de 1 à 9 doivent apparaître sur chaque colonne
- tous les chiffres de 1 à 9 doivent apparaître sur chaque ligne
- tous les chiffres de 1 à 9 doivent apparaître dans chacun des 9 carrés internes.

Les fichiers `sudoku.c` et `sudoku.h` contiennent quelques définitions :

- le type `Grille` est le type des grilles  $9 \times 9$ . On accède aux valeurs avec `G[i][j]`, et on met la valeur `VIDE` (-1) dans les cases vides.
- la fonction `lireGrille` permet de lire une grille (probablement incomplète) au clavier ou dans un fichier. Une grille est rentrée de la manière suivante :

```
.8...16..  
.7.4...21  
5..396...  
2.4.5.13.  
..89.75..  
.57.3.9.2  
...563..9  
31...2.5.  
..58...4.
```

- la fonction `afficheGrille` permet d'afficher une grille (même si elle n'est pas valide).

Si vous mettez la grille dans un fichier `g.txt` vous pouvez éviter de rentrer la grille à la main en utilisant

```
./sudoku < g.txt
```

*Question 1* : programmez la fonction `verifieGrille` qui vérifie si une grille est valide (et complète).

*Question 2* : programmez la fonction `resoudreGrille` qui résout les grilles faciles. (Celles où il ne faut jamais revenir en arrière et qui utilisent un simple raisonnement itératif.)

*Question 3* : programmer une fonction qui résout toutes les grilles de sudoku. (Seulement celles qui sont possibles...)