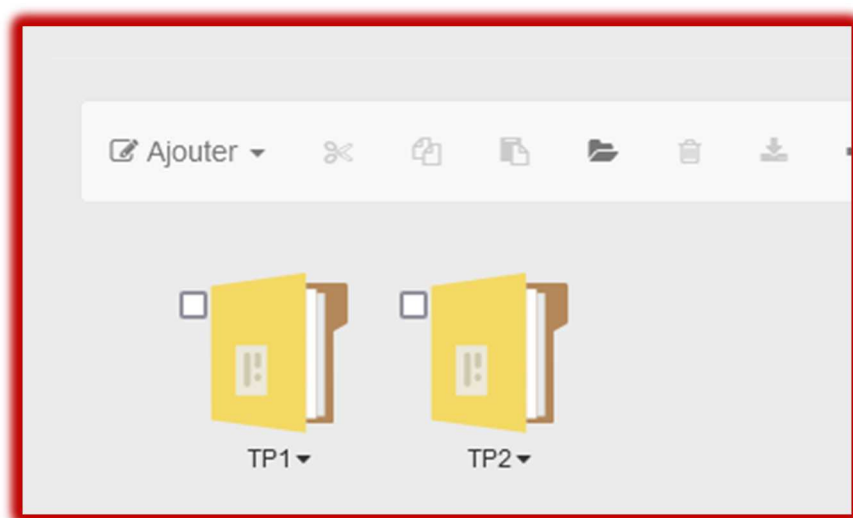


Le travail peut être réalisé en binôme, il est déposé dans le répertoire suivant sous Claroline:



Pour le réaliser, vous devez maîtriser le langage C#. Ce tp s'adresse aux **étudiants issus d'un IUT ou d'un BTS informatique**.

Jeu du Sudoku

Objectifs

Recherche d'un algorithme pour implémenter un jeu.

Objectifs

On souhaite réaliser une grille de sudoku de manière aléatoire

Voici deux grilles générées

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

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

Développement

Vous devez rechercher les structures de données que vous allez utilisées : tableau ou collection

Quelques indications :

Vous pouvez écrire les fonctions suivantes :

```
private boolean nestpasdansligne(int valeur,int lig,int col)
```

// on vérifie que la valeur tirée n'est pas dans la ligne

```
private boolean nestpasdanscolonne(int valeur,int li,int col)
```

// on vérifie que la valeur tirée n'est pas dans la colonne

```
boolean controleinterieurcarre(int v,int ligne, int colonne)
```

// on vérifie que la valeur tirée n'est pas dans le carré

Vous pouvez à présent vous concentrer sur l'algorithme qui va générer votre grille. Lorsque vous générez les premiers nombres, vous recherchez une solution qui appartient à un graphe dont vous devez explorer les nœuds. Cette exploration se fait par une méthode de backtracking, on mémorise nos actions (chemin) et en cas d'échec, on revient en arrière pour rechercher une nouvelle voie mais vous pouvez vous retrouver dans un **circuit**.

Qu'est ce qu'un circuit : Vous êtes sur un nœud mais le nombre trouvé n'est pas bon, vous revenez en arrière pour en chercher un nouveau et vous partez sur une nouvelle **voie** qui vous amène à une mauvaise solution, vous revenez en arrière et vous retombez sur la première voie. Vous êtes alors dans un circuit.

Vous pouvez rechercher une aide sur internet.

Lorsque votre algorithme marche, vous pouvez écrire une interface et si vous avez du temps, écrire le jeu.

Nombre de grilles complètes [Wikipedia]

Il est évident que le nombre de grilles complètes est inférieur au nombre de façons de placer neuf chiffres 1, neuf chiffres 2..., neuf chiffres 9 dans une grille de [81](#) cases. Le nombre de grilles est donc très inférieur à

$$\frac{81!}{9!^9} \approx 5,31306887 \times 10^{70}$$

En effet, dans ce décompte, on ne tient compte d'aucune des contraintes d'unicité.

Le nombre de grilles complètes possibles est également inférieur au nombre de [carrés latins](#) de côté 9.

Enfin, le nombre de grilles complètes possibles est inférieur à $9!^9$ qui correspond au nombre de façons de construire les régions sans tenir compte des contraintes sur les lignes et les colonnes.

En 2005, Bertram Felgenhauer et Frazer Jarvis ont prouvé^[19] que ce nombre de grilles était de :^[20]

$$N = 6\,670\,903\,752\,021\,072\,936\,960 \approx 6,67 \times 10^{21}$$

Ce nombre N est égal à :

$$9! \times 72^2 \times 27 \times 27\,704\,267\,971$$

Problème : choix des cases pour conserver l'unicité de la solution.

Pour ce problème, il semble non résolu à l'heure d'aujourd'hui : si on s'intéresse au nombre de problèmes *proposables*, ce nombre est inconnu ; en revanche, on sait qu'il est nettement plus important que le nombre ~~N~~ indiqué ci-dessus car il existe de très nombreuses façons de présenter des grilles initiales dont la solution (unique) conduit à la même grille terminée (complète). En effet, partant d'une grille complète, on peut construire un problèmeposable par la méthode suivante :

1. Au départ, aucune case de la grille complète n'est « nécessaire ».
2. Choisir une case non « nécessaire » quelconque. Si la suppression de la case choisie conduit à une grille à plusieurs solutions, la marquer comme « nécessaire », sinon la supprimer.
3. Si toutes les cases remplies sont « nécessaires », la grille incomplète est un problèmeposable ; sinon réitérer l'étape précédente.