CIL 2007 • Questionnaire de l'Epreuve de Demi-Finale (sur 100 points)

Problème I – Simplification pittienne — 20 points

A l'école, l'instituteur appelle Pitti au tableau pour simplifier la fraction $\frac{16}{64}$

Nonchalant, Pitti biffe les deux 6 et obtient $\frac{1}{4}$

Horrifié, l'instituteur rend Pitti attentif au fait qu'on ne peut pas simplifier de cette façon; mais, à son étonnement, il doit constater que le résultat est exact.

Problème

Ecrivez un programme qui affiche toutes les fractions à l'écran, une par ligne, qu'on peut simplifier par la méthode "pittienne", mais en livrant un résultat juste. Utiliser le format d'affichage suivant: 16/64

Restrictions

Les numérateurs et dénominateurs se situent chacun dans l'intervalle entier de 10 à 99. Les numérateurs sont toujours strictement inférieurs aux dénominateurs. Pitti sait qu'on ne peut pas simplifier par zéro, les fractions comme 10/20 etc. ne sont donc pas affichées.

Remettez le programme sous le nom SIMPLIF.xxx, avec xxx=PAS ou C(PP). Remettez également le fichier binaire exécutable SIMPLIF.EXE correspondant au programme.

Problème II – Sudoku - solitaires nus, solitaires cachés — 40 points

Sudoku est un jeu de réflexion très à la mode pour l'instant, les règles devraient être connues maintenant.

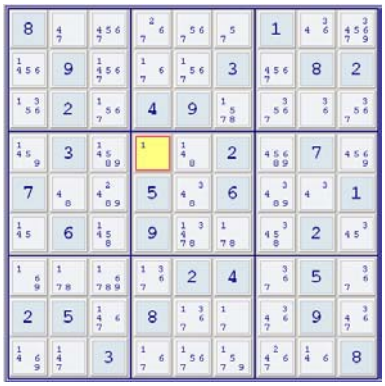
Dans ce problème, nous voulons développer un programme aidant à la résolution en appliquant la stratégie des *solitaires nus* et des *solitaires cachés*. Ces stratégies utilisent la *liste des candidats*.

Etudions un exemple - soit le fichier d'entrée SUDOIN.TXT:

```
8 0 0 0 0 0 1 0 0
0 9 0 0 0 3 0 8 2
0 2 0 4 9 0 0 0 0
0 3 0 0 0 2 0 7 0
7 0 0 5 0 6 0 0 1
0 6 0 9 0 0 0 2 0
0 0 0 0 2 4 0 5 0
2 5 0 8 0 0 0 9 0
0 0 3 0 0 0 0 0 8
```

Les images suivantes ont été créées à l'aide du programme *Simple Sudoku*, par *Angus Johnson*, qu'on peut télécharger gratuitement sur Internet. Ce programme construit la liste des candidats.

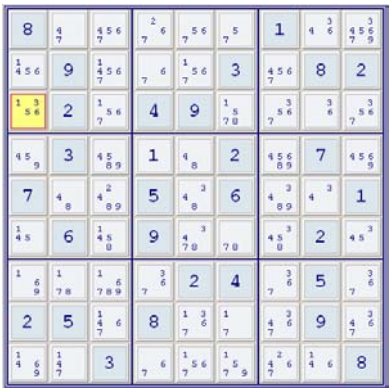
Si l'on examine cette liste, on trouve un solitaire nu, le 1 dans la cellule (4,4).



La case jaune ne contient qu'un seul candidat, le 1. C'est un **solitaire nu**. Cette case doit recevoir le chiffre 1.

Après le placement de ce 1 et la mise à jour de la liste des candidats, on trouve plusieurs solitaires cachés, par exemple le 3 dans la cellule (1,3).

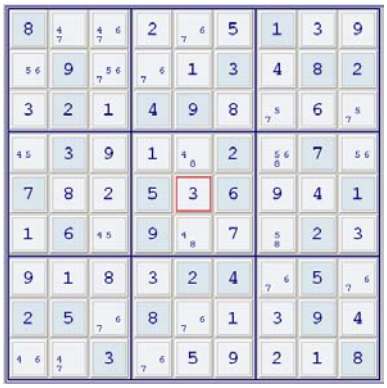
Le premier sous-carré 3x3 ne contient qu'un seul candidat 3 dans la cellule jaune. C'est un **solitaire caché**. Il est caché entre le 1, le 5 et le 6. Cette cellule doit recevoir le chiffre 3.



Les solitaires cachés peuvent apparaître dans des sous-carrés 3x3, mais aussi dans des lignes ou dans des colonnes. La première ligne par exemple, contient un solitaire caché dans la cellule (9,1); le 9. La quatrième colonne, par exemple, contient un solitaire caché dans la cellule (4,7); le 3.

Attention! Après chaque nouveau placement d'un chiffre, d'autres solitaires nus ou cachés peuvent apparaître.

Ici, le placement de tous les solitaires, nus ou cachés, conduit à:



A partir d'ici, on doit appliquer d'autres techniques pour continuer la résolution.

Nous obtenons alors le fichier de sortie SUDOOUT.TXT:

```
8 0 0 2 0 5 1 3 9
0 9 0 0 1 3 4 8 2
3 2 1 4 9 8 0 6 0
0 3 9 1 0 2 0 7 0
7 8 2 5 3 6 9 4 1
1 6 0 9 0 7 0 2 3
9 1 8 3 2 4 0 5 0
2 5 0 8 0 1 3 9 4
0 0 3 0 5 9 2 1 8
```

Problème

Ecrivez un programme qui accepte un Sudoku donné dans le fichier d'entrée SUDOIN.TXT, qui place tous les solitaires nus et cachés et qui écrit le Sudoku obtenu dans le fichier de sortie SUDOOUT.TXT.

Remettez le programme sous le nom SUDOKU.xxx, avec xxx=PAS ou C(PP). Remettez également le fichier binaire exécutable SUDOKU.EXE correspondant au programme.

Problème III - Anagrammes — 40 points

Quand on permute une ou plusieurs lettres d'un mot, on obtient une anagramme.

Exemples

JANE est une anagramme de JEAN. BABA est une anagramme de ABBA.

Problème

Ecrivez un programme qui accepte un mot au clavier et qui affiche toutes les anagrammes à l'écran, une par ligne.

Restrictions

Les mots entrés ne contiennent pas plus de 7 lettres. Toutes les lettres sont des majuscules.

Remettez le programme sous le nom ANAGRA.xxx, avec xxx=PAS ou C(PP). Remettez également le fichier binaire exécutable ANAGRA.EXE correspondant au programme.