

# Programmation d'un solveur en C++ : Kakuro

## 1. Présentation du jeu

Le Kakuro est un jeu de type « nombres fléchés ». Il s'agit d'une grille comportant des cases remplies non modifiables (cases noires), des cases contenant des nombres, des cases vides à remplir avec un nombre et des cases d'indication pouvant contenir deux valeurs, la somme des cases contiguës de la colonne sous la case d'indication et/ou la somme des cases contiguës de la ligne à droite de la case d'indication.

Pour compléter une grille, il s'agit de remplir toutes les cases vides avec des chiffres, de 1 à 9, de telle sorte à ce que les sommes indiquées dans les cases d'indication soient respectées.

Une contrainte supplémentaire est que les colonnes ou lignes contiguës ne peuvent contenir qu'une seule fois chaque nombre de 1 à 9.

Les grilles ont généralement une solution unique.

Ci-dessous la figure 1 comporte un exemple de grille à remplir et la figure 2 contient la grille complétée.

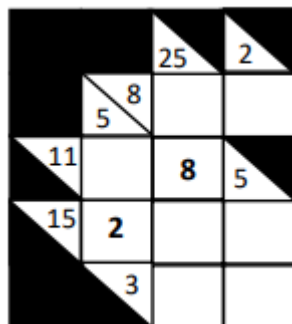


Figure 1 : grille à compléter

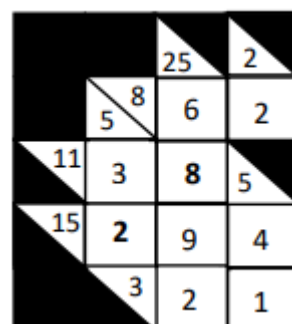


Figure 2 : grille remplie

## 2. Travail demandé

Le travail demandé comportera une implémentation d'un solveur de grille du jeu de Kakuro.

Il sera possible de charger une grille du jeu avec au moins deux formats, dont un format imposé (cf. la section 3). **Afin de rendre le code modulable et extensible, le chargement des formats de grille devra se faire à l'aide du design pattern Factory** (<https://refactoring.guru/design-patterns/factory-method>).

Un solveur de grille devra aussi être disponible. Pour une grille donnée, il sera possible de vérifier qu'il existe au moins une solution, et d'afficher cette solution.

Un rapport court, de 2 à 5 pages, devra accompagner le projet. Ce rapport devra :

- Expliquer vos choix de modélisation et lever les ambiguïtés du sujet.
- Contenir les diagrammes de classes.
- Expliquer les algorithmes principaux sous forme synthétique.
- Présenter la répartition du travail entre étudiants si le travail a été réalisé en binôme.

Implémentation :

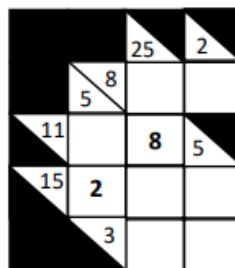
- Votre programme devra utiliser le langage de programmation orientée objet C++.
- L'interface utilisateur est sous forme de ligne de commande.
- Une petite explication de l'utilisation de votre programme est aussi nécessaire, ainsi que tout fichier nécessaire pour le tester.
- Le projet devra contenir un fichier exécutable.

## 3. Les formats de fichiers

### 3.1 Le format imposé

Le format imposé décrivant une grille est le format texte suivant :

```
5 4
# # 25/ 2/
# 5/8 _ _
/11 _ 8 5/
/15 2 _ _
# /3 _ _
```



La première ligne correspond à la dimension de notre Kakuro. Il est, ici, de hauteur 5 (le nombre de lignes) et de largeur 4 (le nombre de colonnes).

# : Case noire

\_ : Case vide à remplir

**B/D** : Case avec somme (B=Somme attendue vers le bas, D=Somme attendue vers la droite)

**Nombres** : Valeur de la case et fixe dans la grille

Vous trouverez sur Moodle quelques exemples de fichiers de Kakuro pour vos tests.

### 3.2 Autres formats

Voici quelques exemples de deuxième format que vous pouvez utiliser au sein de votre application pour charger vos kakuros :

- le format ini
- le format json
- le format yaml

<https://gist.github.com/legolhaas/8ba2abbfa1dd424fe24283151f892ec8>

Vous êtes libres de choisir un de ses formats ou un autre que vous prendrez soin de décrire. Pour vous aider à parser le format ini, json ou yaml vous pourrez vous aider des librairies header-only ci-dessous :

- <https://github.com/jtilly/inih>
- <https://github.com/nlohmann/json> ou <https://github.com/taocpp/json>
- <https://github.com/fktn-k/fkYAML>

Si les formats standards ini, json, yaml ne vous conviennent pas il est également possible de créer un format compact tel que celui-ci :

##3,4,..##10,0,..#7,8,12.#....##9,0,.. et qui pourrait respecter les règles suivantes :

**Cases noires** :

- # pour une case noire sans indice.
- #X,Y, pour une case noire avec indices (ex. #3,4), où :
  - X = somme pour la ligne à droite (0 si absente).
  - Y = somme pour la colonne en dessous (0 si absente).

**Cases à remplir** : . pour une case vide.

**Valeurs fixes** : Chiffres de 1 à 9.

Évidemment, libre à vous d'adapter ces exemples de format pour vous simplifier la tâche.

### 3.3 Chargement via inputs utilisateurs

En lieu et place d'un deuxième format de fichier à charger, il est également envisageable de laisser l'utilisateur renseigner dynamiquement la grille depuis la ligne de commande en lui posant quelques questions qui permettront de remplir la grille au lancement du programme.

Les règles suivantes s'appliquent tout de même :

- ce chargement devra s'intégrer dans le design pattern factory
- vous veillerez à simplifier la tâche de l'utilisateur en rendant le chargement intuitif tout en limitant le nombre de questions.

## 4. Rendu

La remise du projet devra être faite sur moodle, avant le 16 mai à 23h59 sous la forme d'un dossier compressé au format *.zip* ou *.tar.gz* qui inclura :

- le code source du projet
- le binaire exécutable sur le serveur Turing
- un README indiquant de manière simple et concise l'exécution de votre code
- des fichiers de grille de Kakuro tests
- le rapport

La réalisation de ce projet pourra être faite seul ou en binôme. La notation sera adaptée à la taille du groupe.

En cas d'ambiguïté, précisez votre interprétation personnelle du sujet ou envoyez un mail. Toute solution cohérente et justifiée pourra être acceptée.

## 5. Grille de notation

- Modélisation UML: 2 points
- Rapport, explications : 2 point
- Organisation du code (src, bin, resources, etc.) : 1 point
- Code compilable : 1 point
- Rendu d'un exécutable : 1 point
- Qualité du code (écritures des méthodes) : 1,5 point
- Gestion des erreurs / exceptions : 1 point
- Modèle du jeu : 3 points
- Utilisation en ligne de commande : 1 point
- Solveur : 2,5 points
- Utilisation appropriée du Design Pattern Factory : 1 point
- Fonctionnalité unique (**pour les binômes uniquement**) : 3 points. Proposez une propre à votre projet et implémentez là.

Pour les étudiants travaillant seuls, l'évaluation sera plus souple et la fonctionnalité unique n'est pas demandée.