

# Référence du fichier fonctions.h

---

Sudoku. [Plus de détails...](#)

```
#include <stdint.h>
#include <stdio.h>
#include <stdlib.h>
#include <inttypes.h>
#include <math.h>
#include <time.h>
#include <stdbool.h>
#include <string.h>
#include "color.h"
```

[Aller au code source de ce fichier.](#)

## Structures de données

---

struct **caseGrille**

La structure "caseGrille" modelise une case d'une grille de sudoku. On utilise un uint16\_t code sur 16 bits. [Plus de détails...](#)

---

## Macros

---

```
#define COULEUR_BLEU printf(BLUE);
#define COULEUR_VERT printf(GREEN);
#define COULEUR_ORIGINE printf(DEFAULT_COLOR);
#define COULEUR_ROUGE fprintf(stderr, RED);
#define FLUSH_COULEUR_ORIGINE fprintf(stderr, DEFAULT_COLOR);
```

---

## Fonctions

---

size\_t **initTaille** ()

La fonction "initTaille" initialise la taille de la grille de sudoku. [Plus de détails...](#)

---

**caseGrille** \*\* **initGrille** (size\_t taille)

La fonction "initGrille" initialise la grille de sudoku (qui est un tableau a deux dimensions de cases de grille de sudoku), en fonction de la taille

passee en parametre. [Plus de détails...](#)

---

void **afficheGrille** (**caseGrille** \*\*grille, size\_t taille)

La fonction "afficheGrille" affiche la grille de sudoku passee en parametre.

[Plus de détails...](#)

---

int **absentSurLigne** (uint8\_t valeur, **caseGrille** \*\*grille, size\_t ligne, size\_t taille)

---

int **absentSurColonne** (uint8\_t valeur, **caseGrille** \*\*grille, size\_t colonne, size\_t taille)

---

int **absentSurRegion** (uint8\_t valeur, **caseGrille** \*\*grille, size\_t ligne, size\_t colonne, size\_t taille)

---

void **echange** (uint8\_t \*a, uint8\_t \*b)

La fonction "echange" permet d'echanger deux valeurs. [Plus de détails...](#)

---

void **melangerTableauValeurs** (uint8\_t \*tableauValeurs, size\_t taille)

---

bool **genererGrilleComplete** (**caseGrille** \*\*grille, size\_t position, size\_t taille, uint8\_t \*tableauValeurs)

---

void **genererGrilleJeu** (**caseGrille** \*\*grille, size\_t taille, uint8\_t \*tableauValeurs)

---

void **jouer** (**caseGrille** \*\*grille, size\_t taille)

La fonction "jouer" permet de modifier les valeurs associees aux cases de grille. [Plus de détails...](#)

---

int **comparaisonGrilles** (**caseGrille** \*\*premiereGrille, **caseGrille** \*\*deuxiemeGrille, size\_t taille)

La fonction "comparaisonGrilles" permet de comparer deux grilles de sudoku. [Plus de détails...](#)

---

void **sauvegarde** (**caseGrille** \*\*grille, **caseGrille** \*\*grilleSolution, size\_t taille, float temps)

La fonction "sauvegarde" permet de sauvegarder une partie. [Plus de détails...](#)

---

bool **chargerPartie** (size\_t taille, **caseGrille** \*\*grilleSolution, **caseGrille** \*\*grille, float \*temps)

La fonction "chargerPartie" permet de charger une partie. [Plus de détails...](#)

---

void **reinitialiserPartie** (**caseGrille** \*\*grille, size\_t taille)

La fonction "reinitialiserPartie" permet de reinitialiser la grille de jeu comme en debut de partie. [Plus de détails...](#)

---

bool **estInclus** (uint16\_t singleton, uint16\_t possibilitesCase, size\_t taille)

---

bool **crossHatching** (**caseGrille** \*sousGrille, size\_t taille)

---

La fonction "crossHatching" permet d'eliminer un singleton des possibilites des autres cases de sa sous-grille. [Plus de détails...](#)

---

bool **loneNumber** (**caseGrille** \*sousGrille, size\_t taille)

La fonction "loneNumber" est une heuristique qui permet de transformer une aide en singleton si une de ces valeurs n'est pas presente dans les autres cases. [Plus de détails...](#)

---

int **nbValeurs** (unsigned short **aide**)

La fonction "nbValeurs" permet de connaitre le nombre de bits a 1 pour un unsigned short representant le nombre de valeurs d'une liste de possibilites. [Plus de détails...](#)

---

int **nbOccurence** (**caseGrille** \*sousGrille, int indice, size\_t taille)

La fonction "nbOccurence" permet de connaitre le nombre de fois qu'une liste de possibilites apparait. Si elle n'apparait qu'une fois, cela renvoie 0. [Plus de détails...](#)

---

bool **nakedSubset** (**caseGrille** \*sousGrille, size\_t taille)

La fonction "nakedSubset" permet de retirer une liste de possibilite si elle repond au Naked Subset, et de renvoyer un booleen temoignant ou non de l'application de cette regle. [Plus de détails...](#)

---

**caseGrille** \*\* **convertirZoneEnTableau** (**caseGrille** \*\*grille, size\_t ligne, size\_t colonne, size\_t taille)

La fonction "convertirZoneEnTableau" permet de convertir une zone de la grille en tableau. [Plus de détails...](#)

---

**caseGrille** \*\* **convertirColonneEnTableau** (**caseGrille** \*\*grille, size\_t ligne, size\_t colonne, size\_t taille)

La fonction "convertirColonneEnTableau" permet de convertir une colonne de la grille en tableau. [Plus de détails...](#)

---

void **initAide** (**caseGrille** \*\*grille, size\_t taille)

La fonction "initAide" permet d'initialiser l'aide en fonction des valeurs des cases. [Plus de détails...](#)

---

void **aide** (**caseGrille** \*\*grille, const size\_t taille)

La fonction "aide" permet de creer l'aide pour l'ensemble de la grille. [Plus de détails...](#)

---

void **parcoursGrilleAide** (**caseGrille** \*\*grille, size\_t taille, size\_t ligne, size\_t colonne)

La fonction "parcoursGrilleAide" permet d'appliquer les heuristiques sur l'ensemble de la grille. [Plus de détails...](#)

---

void **echangeInt** (int \*a, int \*b)

La fonction "echangeInt" permet d'echanger deux entiers. [Plus de détails...](#)

---

int **insérerPetiteValeur** (int \*T, char noms[10][30], int val, char nom[30])

La fonction "insérerPetiteValeur" permet d'insérer une valeur val si elle est la plus petite parmi tous les elements du tableau T. [Plus de détails...](#)

---

void **demanderNom** (char \*nom)

La fonction "demanderNom" permet de demander et de renvoyer une chaine de caracteres. [Plus de détails...](#)

---

void **score** (int temps, size\_t taille)

La fonction "score" permet de verifier s'il y a un nouveau score. [Plus de détails...](#)

---

void **reinitialiserTop10** (size\_t taille)

La fonction "reinitialiserTop10" permet de reinitialiser le score dans les fichiers Top10\_4.txt, Top10\_9.txt, Top10\_16.txt. [Plus de détails...](#)

---

void **afficherTop10** (size\_t taille)

La fonction "afficherTop10" permet d'afficher le Top 10 d'une taille de grille donnee. [Plus de détails...](#)

---

void **recupeGrille** (**caseGrille** \*\*grille, size\_t taille)

La fonction "recupeGrille" permet de recuperer une grille ecrite dans un fichier. [Plus de détails...](#)

---

void **libereGrille** (**caseGrille** \*\*grille, size\_t taille)

La fonction "libereGrille" permet de liberer en memoire les grilles allouees dynamiquement. [Plus de détails...](#)

---

void **purger** (void)

La fonction "purger" permet de vider le buffer. [Plus de détails...](#)

---

void **clean** (char \*chaine)

La fonction "clean" permet d'eliminer un retour a la ligne a la fin de chaine passee en parametre et vide le buffer. [Plus de détails...](#)

---

size\_t **saisie** (void)

La fonction "saisie" permet de renvoyer la valeur saisie par l'utilisateur. [Plus de détails...](#)

---

## Description détaillée

---

Sudoku.

**Auteur**

Abdoulaye Diallo, Redoine El Ouasti, Simon Galand, Adrien Lamant, Pierre-Louis Latour, Charly Maeder, Pierre Ruffin, Stella Zevio

**Version**

0.2

**Date**

2014-03-26

Sudoku en C (resolution des grilles 4x4, 9x9, 16x16 et jeu).

## Documentation des fonctions

---

```
void afficheGrille ( caseGrille ** grille,  
                   size_t      taille  
                   )
```

La fonction "afficheGrille" affiche la grille de sudoku passee en parametre.

**Auteur**

Abdoulaye Diallo, Stella Zevio

**Paramètres**

**grille** - la grille, qui est un tableau a deux dimensions.

**taille** - la taille de la grille.

```
void afficherTop10 ( size_t taille )
```

La fonction "afficherTop10" permet d'afficher le Top 10 d'une taille de grille donnee.

**Auteur**

Stella Zevio

**Paramètres**

**taille** - la taille de la grille.

```
void aide ( caseGrille ** grille,  
           const size_t  taille  
           )
```

La fonction "aide" permet de creer l'aide pour l'ensemble de la grille.

**Auteur**

Charly Maeder, Pierre Ruffin

**Paramètres**

**grille** - la grille de jeu.

**taille** - la taille de la grille.

```
bool chargerPartie ( size_t      taille,  
                    caseGrille ** grilleSolution,  
                    caseGrille ** grille,  
                    float *      temps  
                    )
```

La fonction "chargerPartie" permet de charger une partie.

**Auteur**

Abdoulaye Diallo, Adrien Lamant

**Paramètres**

**taille** - la taille des grilles.

**grilleSolution** - la solution de la grille de jeu (la grille complete).

**grille** - la grille de jeu.

**temps** - le temps passe a resoudre le sudoku.

**Renvoie**

true si la grille est chargeable

false sinon.

**void clean ( char \* chaine )**

La fonction "clean" permet d'eliminer un retour a la ligne a la fin de chaine passee en parametre et vide le buffer.

**Auteur**

Stella Zevio

**Paramètres**

**chaine** - la chaine passee en parametre.

```
int comparaisonGrilles ( caseGrille ** premiereGrille,  
                        caseGrille ** deuxiemeGrille,  
                        size_t      taille  
                        )
```

La fonction "comparaisonGrilles" permet de comparer deux grilles de sudoku.

**Auteur**

Redoine El Ouasti

**Paramètres**

**premiereGrille** - l'une des grilles que l'on veut comparer.

**deuxiemeGrille** - l'autre grille que l'on veut comparer.

**taille** - la taille des grilles.

**Renvoie**

1 si les grilles sont identiques.

0 si les grilles sont differentes.

```
caseGrille** convertirColonneEnTableau ( caseGrille ** grille,  
                                         size_t      ligne,  
                                         size_t      colonne,  
                                         size_t      taille  
                                         )
```

La fonction "convertirColonneEnTableau" permet de convertir une colonne de la grille en tableau.

**Auteur**

Charly Maeder

**Paramètres**

- grille** - la grille de jeu.
- ligne** - la ligne de la grille où se trouve la case où on veut l'aide.
- colonne** - la colonne de la grille où se trouve la case où on veut l'aide.
- taille** - la taille de la grille.

**Renvoie**

tableau - tableau correspondant a la colonne que l'on voulait convertir.



```
caseGrille** convertirZoneEnTableau ( caseGrille ** grille,  
                                     size_t      ligne,  
                                     size_t      colonne,  
                                     size_t      taille  
                                     )
```

La fonction "convertirZoneEnTableau" permet de convertir une zone de la grille en tableau.

### Auteur

Charly Maeder

### Paramètres

- grille** - la grille de jeu.
- ligne** - la ligne de la grille où se trouve la case où on veut l'aide.
- colonne** - la colonne de la grille où se trouve la case où on veut l'aide.
- taille** - la taille de la grille.

### Renvoie

tableau - tableau correspondant à la zone que l'on voulait convertir.

```
bool crossHatching ( caseGrille * sousGrille,  
                     size_t      taille  
                     )
```

La fonction "crossHatching" permet d'eliminer un singleton des possibilites des autres cases de sa sous-grille.

**Auteur**

Charly Maeder, Pierre Ruffin, Stella Zevio

**Paramètres**

**sousGrille** - la sous-grille a tester.

**taille** - la taille des grilles.

**Renvoie**

true - si cross-hatching s'applique.

false - sinon.

```
void demanderNom ( char * nom )
```

La fonction "demanderNom" permet de demander et de renvoyer une chaine de caracteres.

**Auteur**

Stella Zevio

**Paramètres**

**nom** - Une chaine de caractere.

```
void echange ( uint8_t * a,  
               uint8_t * b  
             )
```

La fonction "echange" permet d'echanger deux valeurs.

**Auteur**

Stella Zevio

**Paramètres**

**a** - la premiere valeur a echanger.

**b** - la deuxieme valeur a echanger.

```
void echangeInt ( int * a,  
                 int * b  
               )
```

La fonction "echangeInt" permet d'echanger deux entiers.

**Auteur**

Stella Zevio

**Paramètres**

**a** - le premier entier que l'on souhaite echanger.

**b** - le deuxieme entier que l'on souhaite echanger.

```
void initAide ( caseGrille ** grille,  
               size_t      taille  
             )
```

La fonction "initAide" permet d'initialiser l'aide en fonction des valeurs des cases.

**Auteur**

Charly Maeder, Pierre Ruffin

**Paramètres**

**grille** - la grille de jeu.

**taille** - la taille de la grille.

```
caseGrille** initGrille ( size_t taille )
```

La fonction "initGrille" initialise la grille de sudoku (qui est un tableau a deux dimensions de cases de grille de sudoku), en fonction de la taille passee en parametre.

**Auteur**

Stella Zevio

**Paramètres**

**taille** - la taille de la grille.

**Renvoie**

grille - la grille de sudoku, qui est un tableau a deux dimensions.

```
size_t initTaille ( void )
```

La fonction "initTaille" initialise la taille de la grille de sudoku.

**Auteur**

Abdoulaye Diallo, Stella Zevio

**Renvoie**

taille - la taille de la grille de sudoku.

```
int insererPetiteValeur ( int * T,  
                        char noms[10][30],  
                        int  val,  
                        char nom[30]  
                        )
```

La fonction "insererPetiteValeur" permet d'insérer une valeur val si elle est la plus petite parmi tous les éléments du tableau T.

**Auteur**

Stella Zevio

**Paramètres**

- T** - un tableau d'entier.
- noms** - Un tableau de char[30].
- val** - Un entier qui est la valeur à insérer.
- nom** - une chaîne de caractères qui est le nom à insérer.

**Renvoie**

- trouve - la position de la valeur à modifier
- 1 - s'il n'y a pas de valeur à modifier

```
void jouer ( caseGrille ** grille,  
            size_t      taille  
            )
```

La fonction "jouer" permet de modifier les valeurs associées aux cases de grille.

**Auteur**

Redoine El Ouasti, Stella Zevio

**Paramètres**

- grille** - la grille de jeu.
- taille** - la taille de la grille.

```
void libereGrille ( caseGrille ** grille,  
                   size_t      taille  
                   )
```

La fonction "libereGrille" permet de liberer en memoire les grilles allouees dynamiquement.

**Auteur**

Redoine El Ouasti, Stella Zevio

**Paramètres**

**grille** - la grille a liberer.

**taille** - La taille de la grille.

```
bool loneNumber ( caseGrille * sousGrille,  
                  size_t      taille  
                  )
```

La fonction "loneNumber" est une heuristique qui permet de transformer une aide en singleton si une de ces valeurs n'est pas presente dans les autres cases.

**Auteur**

Charly Maeder, Pierre Ruffin

**Paramètres**

**sousGrille** - la sous-grille a tester.

**taille** - la taille des grilles.

**Renvoie**

true - si lone-number s'applique.

false - sinon.

```
bool nakedSubset ( caseGrille * sousGrille,  
                  size_t      taille  
                  )
```

La fonction "nakedSubset" permet de retirer une liste de possibilité si elle répond au Naked Subset, et de renvoyer un booléen témoignant ou non de l'application de cette règle.

**Auteur**

Simon Galand, Pierre-Louis Latour

**Paramètres**

**\*sousGrille** - l'ensemble des listes de possibilités.

**taille** - la taille des grilles.

**Renvoie**

true - si naked subset s'applique

false - sinon

```
int nbOccurence ( caseGrille * sousGrille,  
                  int         indice,  
                  size_t      taille  
                  )
```

La fonction "nbOccurence" permet de connaître le nombre de fois qu'une liste de possibilités apparaît. Si elle n'apparaît qu'une fois, cela renvoie 0.

**Auteur**

Simon Galand, Pierre-Louis Latour

**Paramètres**

**\*sousGrille** - l'ensemble de listes de possibilités.

**indice** - l'indice de la case qu'on cherche à dénombrer.

**taille** - la taille des grilles.

**Renvoie**

nbocc - le nombre d'occurrences d'une liste de possibilités.

**int nbValeurs ( unsigned short **aide** )**

La fonction "nbValeurs" permet de connaître le nombre de bits a 1 pour un unsigned short représentant le nombre de valeurs d'une liste de possibilités.

**Auteur**

Simon Galand, Pierre-Louis Latour

**Paramètres**

**aide** - la liste de possibilité a denommer.

**Renvoie**

nbval - le nombre de valeurs d'une liste de possibilités.

```
void parcoursGrilleAide ( caseGrille ** grille,  
                           size_t      taille,  
                           size_t      ligne,  
                           size_t      colonne  
                           )
```

La fonction "parcoursGrilleAide" permet d'appliquer les heuristiques sur l'ensemble de la grille.

**Auteur**

Pierre Ruffin, Charly Maeder

**Paramètres**

**grille** - la grille de jeu.

**taille** - la taille de la grille.

**ligne** - la ligne de la case où on veut l'aide.

**colonne** - la colonne de la case où on veut l'aide.



**void purger ( void )**

La fonction "purger" permet de vider le buffer.

**Auteur**

Stella Zevio

```
void recupeGrille ( caseGrille ** grille,  
                   size_t      taille  
                   )
```

La fonction "recupeGrille" permet de recuperer une grille ecrite dans un fichier.

**Auteur**

Stella Zevio

**Paramètres**

**grille** - la grille que l'on souhaite recuperer.

```
void reinitialiserPartie ( caseGrille ** grille,  
                          size_t      taille  
                          )
```

La fonction "reinitialiserPartie" permet de reinitialiser la grille de jeu comme en debut de partie.

**Auteur**

Pierre Ruffin

**Paramètres**

**grille** - la grille a modifier.

**taille** - la taille des grilles.

**void reinitialiserTop10 ( size\_t **taille** )**

La fonction "reinitialiserTop10" permet de reinitialiser le score dans les fichiers Top10\_4.txt, Top10\_9.txt, Top10\_16.txt.

**Auteur**

Stella Zevio

**Paramètres**

**taille** - la taille de la grille.

**size\_t saisie ( void )**

La fonction "saisie" permet de renvoyer la valeur saisie par l'utilisateur.

**Auteur**

Redoine El Ouasti, Stella Zevio

**Renvoie**

entier - valeur saisie par l'utilisateur.

```
void sauvegarde ( caseGrille ** grille,  
                 caseGrille ** grilleSolution,  
                 size_t      taille,  
                 float       temps  
                 )
```

La fonction "sauvegarde" permet de sauvegarder une partie.

#### Auteur

Abdoulaye Diallo, Adrien Lamant

#### Paramètres

**grille** - la grille de jeu.  
**grilleSolution** - la solution de la grille de jeu (la grille complete).  
**taille** - la taille des grilles.  
**temps** - le temps passe a resoudre le sudoku.

```
void score ( int      temps,  
            size_t    taille  
            )
```

La fonction "score" permet de verifier s'il y a un nouveau score.

#### Auteur

Stella Zevio

#### Paramètres

**temps** - le temps passe a resoudre la grille.  
**taille** - la taille de la grille.