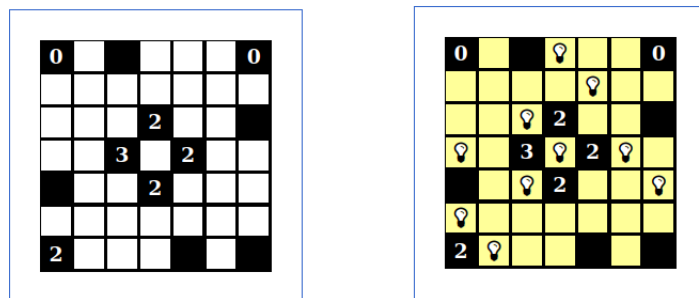


## TP n°5 : Résolution d'un jeu à un joueur.

L'objectif de ce TP est de proposer un algorithme de résolution d'un puzzle de [Light Up](#) selon un algorithme de *backtracking*.



Un jeu sera chargé depuis un des fichiers `levelX.txt` depuis la fonction `main`, et sera ensuite représenté par une structure appelée `board`. On pourra également utiliser une structure `state` pour stocker le vecteur des positions des lampes qui sont déjà dans le tableau. Ces structures sont définies dans le fichier `lightup.c`.

Il faudra ensuite définir les fonctions suivantes, pour lesquelles un prototype est proposé:

1. `void print_board (board b);` qui affiche le tableau de jeu;
2. `board copy_board (board b);` qui réalise une copie de l'état actuel du board, pour éventuellement revenir en arrière dans la résolution;
3. `void printLights` qui affiche le vecteur courant de position des ampoules;
4. `void add_light (state *s,int pos);` qui rajoute une lumière supplémentaire dans le tableau de jeu et dans le vecteur d'états;
5. `void fill_board (board *b, state *s);` qui renvoie le tableau avec toutes les cases éclairées en fonction des positions de lampes;
6. `int check (board *b, state *s);` qui vérifie si l'état actuel des lampes respecte les contraintes du jeu, et renvoie 1 si oui, 0 sinon;
7. `int is_solved(board *b, state *s)` qui détermine si le jeu est résolu, c'est-à-dire que les contraintes sont respectées et que toutes les cases blanches du tableau sont des lampes ou éclairées;
8. `int solve (board *b,state *s);` qui résout le jeu par une méthode de *backtracking*.

Mesurez ensuite l'efficacité de cet algorithme de résolution sur les différents niveaux présentés, et comparer le temps d'exécution avec l'algorithme de Chiu, Chou, Yang et Yen que vous trouverez dans le fichier `ccyy.c`.