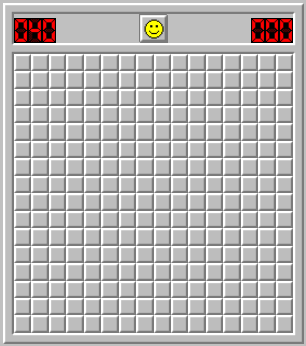
TP1

Minesweeper

## Règles de jeu

Minesweeper se joue sur une grille où l’on a placé des mines (bombes) aléatoirement.



Le but est de dévoiler toutes les cellules qui ne contiennent pas de bombe. Lorsqu’on clique sur une cellule qui ne contient pas de bombe, un nombre apparaît, indiquant le nombre de bombes qui se trouvent dans les cellules adjacentes.



Par déduction logique, on peut déterminer quelles cellules contiennent forcément des bombes et, par conséquent, celles qui n’en contiennent pas.

Attention, lorsqu’on clique sur une bombe, on perd la partie.



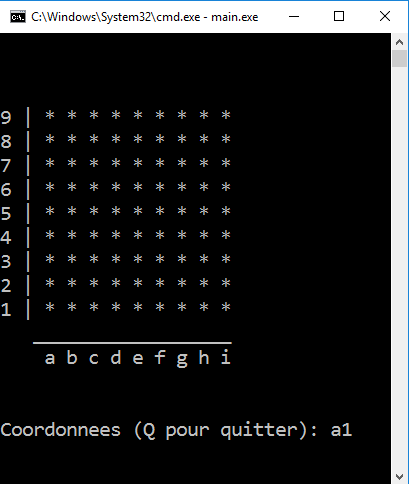
Vous remarquez peut-être que certaines cellules ne contiennent ni nombre, ni bombe. Nommons-les des cellules vides. La particularité d’une cellule vide est qu’aucune bombe ne lui est adjacente. Lorsqu’on la clique, toutes les cellules adjacentes qui ne contiennent pas de bombe sont dévoilées. Et si une cellule vide figure parmi ces cellules adjacentes, toutes les cellules adjacentes à cette nouvelle cellule vide seront dévoilées si elles ne contiennent pas de bombe, et ainsi de suite. On applique donc le même traitement aux cellules adjacentes qu’à la cellule initiale : récursivité !

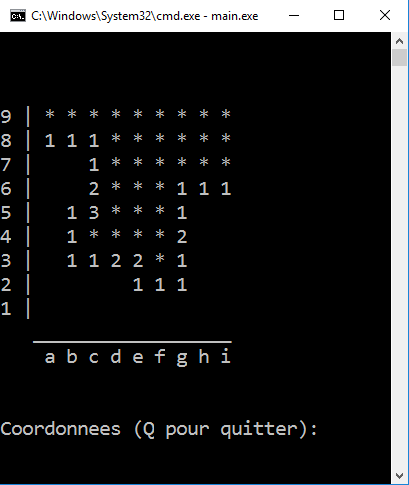
Analyse :

* Comment peut-on diviser le problème en de plus petits problèmes du même type ?
  + On traite l’ensemble des cellules adjacentes individuellement comme la cellule initiale.
* Comment est-ce que chaque appel récursif diminue la taille du problème ?
  + On passe de (potentiellement) 9 cellules à une.
* Quelle(s) instance(s) de ces sous-problèmes peu(ven)t servir de cas de base ?
  + Si on tombe sur une cellule non vide et cachée on arrête.
    - Si ce n’est pas une bombe, on la dévoile.
  + Si on tombe sur une cellule dévoilée, on arrête.
  + Si on tombe sur la bordure de la grille, on arrête.
* Alors que la taille du problème diminue, arrivera-t-on au cas de base ?
  + Il y a un nombre fini de cellules sur la grille, donc oui.

## Votre tâche

Vous devez faire une version ASCII de Minesweeper. Vous présentez la grille à l’utilisateur et l’utilisateur entre les coordonnées de la cellule qu’il désire dévoiler. Vous présentez la nouvelle grille et ça recommence jusqu’à ce que l’utilisateur gagne, perde ou désire quitter.





Lorsqu’une partie est terminée, vous devez demander à l’utilisateur s’il veut recommencer. Sinon, on quitte.

Je vous fournis du code de base où :

* Il y a deux librairie inclues.
  + stdio et stdlib
    - Souvenez-vous que le placement des bombes est aléatoire.
      * Vous aurez peut-être besoin d’autres librairies.
* Il y a plusieurs constantes :
  + La taille de la grille est fixe (9X9).
  + Le nombre de bombes est fixe (10).
  + La valeur d’une bombe est fixe.
  + Les caractères utilisés sont fixes.
* Une déclaration de structure :
  + Un membre pour la valeur de la cellule.
    - Soit que c’est une bombe ou bien le nombre de bombes qui y sont adjacentes.
  + Un membre pour l’état de la cellule, déterminant si elle est cachée ou non.
* 3 fonctions :
  + void initGrille(struct cellule grille[][MAXTAILLE])
    - Cette fonction initialise la grille entière de cellules vides.
  + void imprimerGrille(struct cellule grille[][MAXTAILLE])
    - Cette fonction imprime la grille.
  + int main(int argc, char \*\*argv)
    - Pour l’instant le main(…) ne fait qu’afficher une grille initialisée de cellules vides.

Vous aurez à faire :

* Une boucle d’événement où le jeu se déroule.
* Une fonction récursive pour le dévoilement des cellules.
* Une mise en œuvre qui assure que la première cellule sélectionnée ne contienne pas de bombe.
  + Simple avec deux initialisations séparées.
    - Une qui met des cellules vides (celle que j’ai fournie).
    - Une autre qui sème les bombes aléatoirement et parcourt la grille pour ajuster les valeurs des cellules en conséquence.
      * Ceci se fait donc après avoir reçu les premières coordonnées de l’utilisateur.

## Remise

Vous devez remettre le fichier contenant uniquement le code source de votre jeu. Vous pouvez remettre le projet à tout moment d’ici le 26 septembre, avant la fin du cours.