

L'objectif de ces TP est de reprendre contact avec les fonctions de base en Python : manipulation de listes, de chaînes de caractères, et de matrices ; lecture de fichiers.

1 Lièvres et renards

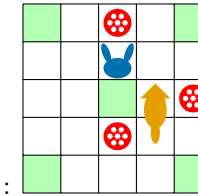
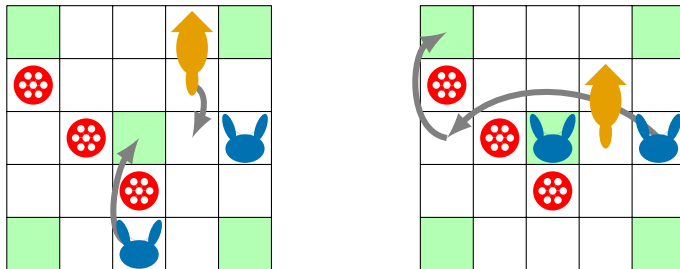
Le jeu Lièvres et renards est un casse-tête commercialisé par la société Smart Games, et se joue sur une grille de 5 lignes et 5 colonnes. Dans cette grille peuvent se trouver :

- des champignons (obstacles inamovibles qui occupent une case, représentés par ) ,
- des lièvres (qui se déplacent et occupent une case, représentés par ) ,
- des renards (qui se déplacent, occupent deux cases contigües, représentés par ) .

On joue en plaçant les différents objets en suivant un modèle, puis en déplaçant les lièvres et les renards de manière à amener tous les lièvres dans des cases *terrier* (soit au centre, soit dans un coin). Plus précisément :

- Les renards peuvent se déplacer uniquement d'une case vers l'avant ou l'arrière, à condition d'arriver sur une case vide.
- Les lièvres ne peuvent se déplacer qu'en sautant par-dessus un ou plusieurs obstacles alignés (et arrivent, dans ce cas, dans la première case vide après le/les obstacles).
- Les lièvres peuvent servir d'obstacle à d'autres lièvres.
- Un lièvre peut sauter par-dessus un renard en longueur ou sur le côté.

La figure suivante illustre comment on peut gagner une situation donnée (en 4 coups, ici).



► **Question 1** Résoudre le plateau suivant :

► **Question 2** On va numéroté les cases de 0 à 24 dans l'ordre des lignes, pour simplifier la représentation d'un plateau. Un plateau est donné par la liste de trois choses :

- une liste d'entiers (les cases contenant un lapin),
- une liste de tuples de deux valeurs : la case contenant la tête du renard, et 0 si le renard est horizontal, 1 s'il est vertical.
- une liste d'entiers (les cases contenant un champignon),

Ainsi, le plateau donné en introduction serait :

```
plateau = [ [14, 22],
             [(3,1)],
             [5, 11, 17],
             ]
```

Dans toute la suite, on ne considérera que des plateaux valides, ce qu'on ne vérifiera jamais. Écrire une fonction `gagne(plateau)` qui renvoie `True` si le plateau est gagné (c'est-à-dire si tous les lièvres sont dans un terrier).

► **Question 3** Un plateau peut aussi utilement se représenter à l'aide d'une matrice 5×5 d'entiers en utilisant la convention suivante :

- 0 désigne une case vide,
- 1 désigne une case contenant un lièvre,
- 2 désigne une case contenant un champignon,
- 3 désigne une case contenant un renard.

Écrire une fonction `plateau_to_grille(plateau)` qui renvoie la matrice associée à `plateau`, puis une fonction `print_grille(grille)` qui affiche une grille (on essaiera de rendre le résultat lisible).

2 Coups possibles

Dans un premier temps, il va nous être indispensable de déterminer, à partir d'un plateau, les plateaux *accessibles en un coup* depuis ce plateau. Si vous vous sentez, vous pouvez simplement sauter à la question 6. Les deux questions intermédiaires facilitent le travail.

► **Question 4** Écrire une fonction `coups_du_renard(grille,renard)` qui prend en argument une grille (donc une matrice 5×5) et un renard (par exemple (3,1)) et renvoie la liste des « renards possibles » au coup d'après (il faut vérifier que le renard ne dépasse pas de la grille, et que la nouvelle case d'arrivée est libre. Pour rappel, un renard ne peut se déplacer qu'en avant ou en arrière).

► **Question 5** Écrire une fonction `coups_du_lapin(grille,i)` qui prend en argument une grille et la case d'un lièvre *i*, et renvoie la liste des positions vers lesquelles pourrait sauter un lièvre qui serait en case *i* (on ne vérifiera pas qu'il s'y trouve effectivement). Pour rappel, un lièvre ne peut pas se déplacer sur une case adjacente. La liste des résultats contiendra donc entre 0 et 4 valeurs.

► **Question 6** Écrire une fonction `coups_possibles(plateau)` qui prend en argument un plateau et renvoie une liste de plateaux possibles après un coup.

3 Résolution

Pour résoudre le problème, on ne va rien faire de très subtil : on va essayer tous les mouvements possibles depuis toutes les configurations atteignables. On traitera les plateaux par ordre croissant de nombre de coups passés depuis la situation initiale.

► **Question 7** Expliquer l'idonéité¹ d'une file dans cette situation. Ré-écrire les fonctions `nouvelle_file()`, `push(file,x)`, `pop(file)` et `est_vide(file)`.

► **Question 8** Écrire une fonction `resolution(plateau)` qui résout le problème posé par `plateau` et renvoie la situation finale. On créera pendant cette fonction une liste des plateaux atteints afin de ne jamais placer dans la file un plateau qui a déjà été traité. On pourra renvoyer un message particulier si aucune solution n'est atteignable.

¹C'est bien un mot, et il est très joli.

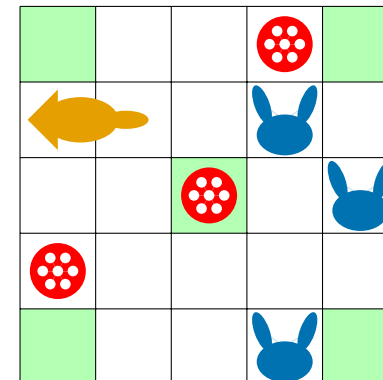
4 Tout résoudre

► **Question 9** Dans le manuel, Lièvres et renards propose 60 problèmes. Par chance, ces problèmes sont disponibles dans le fichier `/home/users/etourniaire/Documents/TP/01-lievres.txt`. Le format est le suivant :

- chaque ligne du fichier décrit un problème,
- chaque ligne est découpée en plusieurs parties suivant des points-virgules, et chaque partie est de la forme :
 - `nXX` pour le puzzle numéro `XX`
 - `mXX` si le manuel indique que le nombre de coups minimal pour résoudre le puzzle est de `XX`.
 - `cXX` s'il y a un champignon dans la case `XX` (entre 0 et 24).
 - `lXX` s'il y a un lièvre dans la case `XX`
 - `rXX` s'il y a un renard horizontal dans les cases `XX` et `XX + 1`
 - `qXX` s'il y a un renard vertical dans les cases `XX` et `XX + 5`

Résoudre tous les puzzles, et trouver ceux pour lesquels le manuel se trompe sur le nombre minimum de coups.

► **Question 10** [Difficile ?²] Concevoir un ou des puzzles dont la longueur de résolution est maximale.



²Je n'ai sincèrement aucune idée de la difficulté de cette question.