

Bases de la programmation impérative

Ensimag 1A

1. Heure à la seconde suivante

Écrire une fonction `heure_seconde_suivante(heure, minutes, secondes)` renvoyant l'heure obtenue en incrémentant d'une seconde le temps donné en paramètre.

2. Carrés

Écrire une fonction `est_carre(points)` prenant en argument un tableau de points formant un quadrilatère (les points sont donc ordonnés) et renvoyant s'ils forment un carré.

Pour rappel, un carré peut être caractérisé de nombreuses façons, par exemple avec uniquement des tests d'orthogonalité (il en faut 4), ou avec uniquement des tests d'égalité de longueur (4 également), ou encore avec un test d'égalité de vecteurs et deux tests d'orthogonalité.

3. Échecs

On considère la programmation d'un jeu d'échecs.

L'échiquier est codé par un tableau à deux dimensions, chaque case étant identifiée par deux indices entiers compris entre 0 et 7. Chaque case contient `None` si elle est vide ou un tuple formé d'un booléen codant la couleur de la pièce présente ainsi qu'un caractère codant le type de pièce. On prendra les blancs à vrai, les noirs à faux ainsi que les lettres suivantes : r pour roi, d pour dame, t pour tour, c pour cavalier, f pour fou, p pour pion.

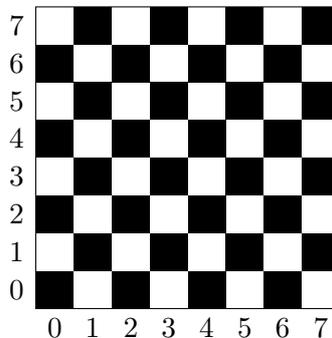


FIGURE 1 – Échiquier

3.1 Représentation

3.1. si `echiquier` est la variable contenant l'échiquier, comment désigner la couleur de la pièce ligne 5, colonne 3 ?

3.2. mettez un fou blanc sur la case (2,5)

- 3.3. dessinez un échiquier avec quelques pièces ainsi que sa représentation en mémoire
- 3.4. proposez d'autres manières de coder un échiquier

3.2 Mouvements

On cherche à programmer une fonction qui vérifie si un coup, identifié par une position de départ (2 entiers) ainsi qu'une position d'arrivée (2 autres entiers) est valide.

- 3.1. Donnez l'implémentation de la fonction `destination_valide(echiquier, x, y)` renvoyant un booléen indiquant si les coordonnées (x, y) désignent une case valide (dans l'échiquier) et inoccupée.
- 3.2. Donnez l'implémentation d'une fonction vérifiant si le déplacement d'une tour est valide (on supposera toutes les cases entre le départ et la destination inoccupées)
- 3.3. Donnez l'implémentation d'une fonction vérifiant si le déplacement d'un fou est valide (on supposera toutes les cases entre le départ et la destination inoccupées)
- 3.4. Donnez l'implémentation d'une fonction vérifiant si le déplacement d'un cheval est valide.
- 3.5. Sans vous fatiguer (i.e., en réutilisant les fonctions déjà écrites), donnez l'implémentation d'une fonction vérifiant si le déplacement d'une reine est valide.