

TP informatique

Programmation du jeu Mastermind

Le Mastermind est un jeu dans lequel le joueur doit deviner une combinaison secrète, qui est une liste de 4 chiffres différents (ce sont des couleurs dans le jeu original, mais on utilisera des chiffres commençant à partir de 0 pour simplifier).

Pour cela, le joueur peut proposer une solution et l'autre joueur (ici, l'ordinateur) lui indique le nombre de chiffres bien placés et le nombre de chiffres mal placés par rapport à la combinaison secrète.

Par exemple, si la combinaison secrète est [0, 2, 4, 3] et que le joueur propose [0, 1, 3, 2], il obtiendra comme réponse : 1 bien placé (le 0) et 2 mal placés (le 2 et le 3).

1. Écrire une fonction `bien_place` telle que `bien_place(L1, L2)` renvoie le nombre d'éléments de L1 qui sont bien placés dans L2. Par exemple, `bien_place([0, 2, 4, 3], [0, 1, 3, 2])` doit renvoyer 1 (le 0 est placé au même endroit dans les deux listes).
On supposera que les deux listes L1 et L2 sont de même taille.
2. Écrire une fonction `appartient` telle que `appartient(e, L)` détermine si l'élément `e` est dans la liste L.
3. Écrire une fonction `mal_place` telle que `mal_place(L1, L2)` renvoie le nombre d'éléments de L1 qui sont mal placés dans L2. Par exemple, `mal_place([0, 2, 4, 3], [0, 1, 3, 2])` doit renvoyer 2 (le 3 est dans les deux listes à des positions différentes, ainsi que 2).
On supposera que les deux listes L1 et L2 sont de même taille.

La combinaison secrète sera générée aléatoirement par l'ordinateur. Pour cela on utilisera le module `random` en écrivant `from random import *` qui permet d'utiliser la fonction `randint` telle que `randint(a, b)` renvoie un entier aléatoire entre `a` et `b` inclus.

4. Écrire une fonction `generer` telle que `generer(n)` renvoie une liste de taille 4 contenant des entiers aléatoires **différents** entre 0 et `n`.
Indice : on pourra choisir un nouvel entier aléatoire tant que celui-ci appartient à la liste des chiffres déjà choisis.

On veut maintenant que le joueur puisse entrer chacune de ses propositions. Pour cela on utilisera `int(input())` qui demande un chiffre à l'utilisateur et renvoie sa valeur. On peut alors utiliser ce résultat comme un entier classique, par exemple en écrivant `a = int(input())` pour le stocker dans une variable `a`.

5. Écrire une fonction `proposition` telle que `proposition()` demande 4 chiffres à l'utilisateur et renvoie la liste les contenant.
Remarque : `proposition` est une fonction sans argument, d'où les parenthèses vides.
6. Écrire une fonction `jeu` telle que `jeu(n)` génère une combinaison secrète en utilisant des chiffres entres 0 et `n`, puis demande une proposition au joueur tant que le joueur n'a pas trouvé la combinaison secrète, en affichant à chaque fois le nombre de chiffres bien placés/mal placés.
7. Modifier la fonction précédente pour afficher le nombre de propositions qui ont été effectuées avant de trouver la solution.
8. Faire en sorte de pouvoir jouer avec des éventuelles répétitions dans la combinaison secrète (il faut changer `generer` mais aussi `bien_place` et `mal_place`).