

# Test algorithme fin

Sujet: Jeu de mastermind

<https://fr.wikipedia.org/wiki/Mastermind>

## Principe

Un joueur commence par placer son choix de pions sans qu'ils soient vus de l'autre joueur à l'arrière d'un cache qui les masquera à la vue de celui-ci jusqu'à la fin de la manche.

Le joueur qui n'a pas sélectionné les pions doit trouver quels sont les quatre pions, c'est-à-dire leurs couleurs et positions.

Pour cela, à chaque tour, le joueur doit se servir de pions pour remplir une rangée selon l'idée qu'il se fait des pions dissimulés.

Une fois les pions placés, l'autre joueur indique :

1. le nombre de pions de la bonne couleur bien placés en utilisant le même nombre de pions rouges ;
2. le nombre de pions de la bonne couleur, mais mal placés, avec les pions blancs.

Il arrive donc surtout en début de partie qu'il ne fasse rien concrètement et qu'il n'ait à dire qu'aucun pion ne correspond, en couleur ou en couleur et position.

La tactique du joueur actif consiste à sélectionner en fonction des coups précédents, couleurs et positions, de manière à obtenir le maximum d'informations de la réponse du partenaire puisque le nombre de propositions est limité par le nombre de rangées de trous du jeu. Dans la plupart des cas, il s'efforce de se rapprocher le plus possible de la solution, compte tenu des réponses précédentes, mais il peut aussi former une combinaison dans le seul but de vérifier une partie des conclusions des coups précédents et de faire en conséquence la proposition la plus propice à la déduction d'une nouvelle information.

Le joueur gagne cette manche s'il donne la bonne combinaison de pions sur la dernière rangée ou avant. Dans tous les cas, c'est à son tour de choisir les pions à découvrir.

## Adaptation à l'informatique

Vous remplacez les couleurs de pions par des chiffres de 0 à 9.

Le joueur joue contre l'ordinateur qui génère le code.

Au début d'une partie, l'ordinateur génère 4 chiffres aléatoires à deviner.

Le joueur dispose de 10 essais par partie.

Pour chaque essai, le joueur entre 4 chiffres.

Le programme lui retourne le nombre de chiffres bien placés et le nombre de chiffres mal placés.

En fin de partie, on demande au joueur s'il souhaite rejouer ou quitter le jeu.

1. Créer une structure pour le jeu qui contient la combinaison à deviner, le nombre d'essais restants du joueur et le nombre de parties gagnées et perdues. (1 points)
2. Au début d'une partie, l'ordinateur génère aléatoirement la combinaison secrète et on initialise le nombre d'essais restants.  
Cette combinaison est stockée dans un tableau de la structure. (1 point)
3. Le joueur entre successivement les 4 chiffres de la combinaison (1 point)
4. Le programme teste la combinaison à l'aide d'une fonction et renvoie le nombre de chiffres bien placés et ceux présents dans la combinaison mais mal placés, dans une structure. (4 points)
5. Si le joueur n'a plus d'essais restants, la partie s'arrête (0,5 point)
6. Si le joueur trouve la bonne combinaison, la partie est gagnée, elle s'arrête. (0,5 point)
7. On informe le joueur de son nombre de parties gagnées et perdues (0,5 point)
8. On demande au joueur s'il souhaite commencer une nouvelle partie ou non. (0,5 point)
9. Vous fournissez un repository git public de votre projet. Pas d'invitation (1 point)

Qualité du code: (10 points)

- Utilisez des fonctions
- Vos variables doivent être bien nommées
- Votre indentation doit être propre
- Votre code doit être clair et commenté



