UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

Licence 1 INFO
INFO0101

TP n°7

Esprit de maître

1 Mastermind

Le but du Mastermind est de deviner dans l'ordre une suite de chiffres (de 0 à 9) générée aléatoirement par l'ordinateur (dans la version originale, il s'agit de retrouver des couleurs). Lorsque le joueur trouve la suite dans le même ordre, il a gagné. Pour cela, à chaque tour, le joueur fait une proposition et on lui indique les chiffres qui sont bien placés (à la bonne position) et ceux qui sont mal placés (présents dans la solution mais pas à la même position).

Pour représenter la solution à trouver et la proposition du joueur, nous choisissons d'utiliser un tableau d'entiers dont la taille est fixée (à 4 pour commencer). Pour le résultat, nous utilisons un tableau de caractères dans lequel nous plaçons des 'B' pour bien placé, 'M' pour mal placé et un espace sinon. Par exemple, si la solution à trouver est [6, 4, 3, 0], voici un exemple de déroulement de partie :

- L'utilisateur propose [1, 2, 3, 4], le résultat est [' ', ' ', 'B', 'M'].
- L'utilisateur propose [4, 5, 3, 6], le résultat est ['M', ' ', 'B', 'M'].
- L'utilisateur propose [6, 4, 3, 7], le résultat est ['B', 'B', 'B', ' '].
- *etc.*

L'application sera réalisée à l'aide de plusieurs fonctions/procédures (9 au total). Aussi, il est vivement conseillé pour chaque question d'écrire la fonction/procédure demandée et de la tester dans le `main` avant de passer à la question suivante.

1. Écrivez la procédure `afficherTableauEntiers` qui permet d'afficher un tableau d'entiers passé en paramètre en séparant chaque chiffre par un espace.

```
public static void afficherTableauEntiers(int[] t)
```

2. Écrivez la procédure `afficherTableauCaracteres` qui permet d'afficher un tableau de caractères passé en paramètre en séparant chaque caractère par un espace.

```
public static void afficherTableauCaracteres(char[] t)
```

3. Écrivez la fonction `remplirSolution` qui crée un tableau dont la taille est spécifiée en paramètre et le remplit aléatoirement de chiffres entre 0 et 9.

```
public static int[] remplirSolution(int taille)
```

4. Écrivez la fonction `saisirProposition`. Elle demande à l'utilisateur de saisir une proposition qui est retournée sous la forme d'un tableau d'entiers. Chaque chiffre saisi doit forcément être dans l'intervalle [0, 9].

```
public static int[] saisirProposition(int taille)
```

5. Écrivez la fonction `bienPlacee` qui retourne true si la valeur `valeur` est bien située à la position `position` dans la solution.

```
public static boolean bienPlacee(int[] solution, int valeur, int
   position)
```

6. Écrivez la fonction `malPlacee` qui retourne `true` si la valeur `valeur` est bien située dans la solution mais pas à la position `position`.

```
public static boolean malPlacee(int[] solution, int valeur, int
   position)
```

7. Écrivez la fonction `compare`. Elle permet de comparer la proposition de l'utilisateur avec la solution à trouver et retourne un tableau de caractères qui contient des 'B' pour les valeurs bien placées, 'M' pour celles mal placées et un espace sinon. Cette fonction doit utiliser les fonctions `bienPlacee` et `malPlacee`.

```
public static char[] compare(int[] proposition, int[] solution)
```

8. Écrivez la fonction `gagne` qui prend en paramètre le résultat et indique si l'utilisateur a gagné. Indication : un utilisateur a gagné s'il n'y a que des 'B' dans le résultat.

```
public static boolean gagne(char[] resultat)
```

9. Écrivez la fonction `afficherResultat`. Elle prend en paramètre la proposition de l'utilisateur ainsi que la solution à trouver. Elle affiche la proposition à l'écran ainsi que le résultat de la comparaison (utilisez la fonction `compare`). Elle retourne `true` si l'utilisateur a gagné (en utilisant la fonction `gagne`).

```
public static boolean afficherResultat(int[] proposition, int[]
   solution)
```

10. Écrivez le `main` qui réalise les actions suivantes :
    - Génération d'une combinaison aléatoire.
    - Demande à l'utilisateur de saisir des propositions jusqu'à ce qu'il gagne.
    - Affichage du nombre de tours qu'il a été nécessaire pour trouver la solution.

2 Mastermind amélioré

Cet exercice est facultatif.

Pour simplifier le jeu, on n'autorise pas qu'il y ait plusieurs fois la même valeur dans la solution.

1. Proposez une modification pour la fonction `remplirSolution`.
2. Pour apporter plus de souplesse, on demande à l'utilisateur de choisir une taille pour la solution, ainsi qu'un intervalle de valeurs (au lieu de 0 à 9).

Normalement, c'est seulement le nombre de chiffres bien placés et le nombre de chiffres mal placés qui sont indiqués.

3. Modifiez votre programme pour que la fonction compare retourne un tableau de taille quelconque qui ne contient que des 'B' et des 'M'. Si aucun pion n'est présent dans la solution, la fonction retourne `null`.
4. Proposez un jeu à deux joueurs où c'est le premier joueur qui propose une combinaison et où le deuxième joueur tente de trouver. Ensuite, c'est l'inverse. Le gagnant est celui qui gagne en un minimum de coups.