

TP 5 bis : tableaux

Informatique fondamentale (IF1)

Semaine du 3 décembre 2007

1 Tableaux

Exercice 1 Dans la méthode `main`, commencez par déclarer un tableau, demandez à l'utilisateur la taille souhaitée du tableau, puis les éléments du tableau.

Pour chaque méthode écrite dans la suite, n'oubliez pas de tester la compilation et le résultat dans la méthode `main`.

Exercice 2 Écrivez une méthode `void affiche(int tab[])` qui prend en argument un tableau d'entiers `tab` et qui affiche les éléments de ce tableau.

Exercice 3 Écrivez une méthode `int somme(int tab[])` qui prend en argument un tableau d'entiers `tab` et qui renvoie la somme des nombres qu'il contient.

Exercice 4 Écrivez une méthode `int minimum(int tab[])` qui prend en argument un tableau d'entiers `tab` et qui renvoie le plus petit des éléments de ce tableau.

Exercice 5 Écrivez une méthode `int maximum(int tab[])` qui prend en argument un tableau d'entiers `tab` et qui renvoie le plus grand des éléments de ce tableau.

Exercice 6 En utilisant la méthode `somme()` précédente, écrivez une méthode qui renvoie la moyenne des éléments du tableau `tab`.

Exercice 7 Écrivez une méthode `boolean est_dedans(int tab[], int element)` qui prend en argument un tableau d'entiers `tab` et qui renvoie `true` si l'entier `element` est compris dans le tableau `tab` et `false` sinon.

Exercice 8 Écrivez une méthode `int cherche(int tab[], int element)` qui prend en argument un tableau d'entiers `tab` et qui renvoie le premier indice `i` tel que `tab[i] = element` (elle renverra `-1` si le tableau ne contient pas l'entier `element`).

Exercice 9 Écrivez une méthode `boolean est_trie(int tab[])` qui teste si le tableau `tab` est trié dans l'ordre croissant.

Exercice 10 Écrivez une méthode `void inverse_element(int tab[], int i, int j)` qui inverse les éléments `i` et `j` du tableau.

2 MasterMind

Vous allez programmer un jeu de Mastermind, où l'utilisateur doit deviner une combinaison générée aléatoirement par l'ordinateur. Le joueur dispose d'un nombre déterminé d'essais. A chaque essai d'une combinaison, l'ordinateur fournit deux indices : le nombre de pions correctement placés et le nombre de pions de la bonne couleur mais incorrectement positionnés.

Vous prendrez ici une combinaison de 5 pions de 4 couleurs différentes. La couleur d'un pion sera représentée par un entier compris entre 0 et 3. Pour une combinaison de 5 pions, nous allons donc utiliser un tableau de 5 entiers.

*utilisez la fonction `Math.random()` qui renvoie un **double** compris entre 0 et 1*

Exercice 11 Commencez par écrire un programme qui choisit une combinaison au hasard et qui demande une proposition au joueur.

Exercice 12 Le programme doit maintenant compter le nombre de "bien placés" au joueur et demander une nouvelle proposition tant que le joueur n'a pas trouvé la bonne combinaison.

Exercice 13 Rajoutez la prise en compte des "mal placés". Exemple : la proposition est : "1 1 0 2 3" et la solution : "1 0 0 1 3", le programme doit répondre "3 bien placés, 1 mal placé".

Exercice 14 Rajoutez un compteur, pour comptez le nombre de coups du joueur. Limiter le nombre de coups à 15 (passé ce nombre, le joueur a perdu).