

R1.01 - Initiation au développement

Sujet: mélanger + inverser les tableaux

Instructeur: Joseph AZAR - joseph.azar@univ-fcomte.fr

Problème:

L'algorithme doit simuler un jeu appelé *reversi*. Le jeu commence par afficher les chiffres de 1 à 9 dans le désordre. Le joueur donne un entier compris entre 1 et 9 qui indique le nombre de chiffres à inverser à partir de la gauche. La nouvelle configuration du jeu est affichée. Par exemple pour la configuration :

5 4 6 2 1 7 9 8 3

si le joueur donne l'entier 5, la nouvelle configuration sera:

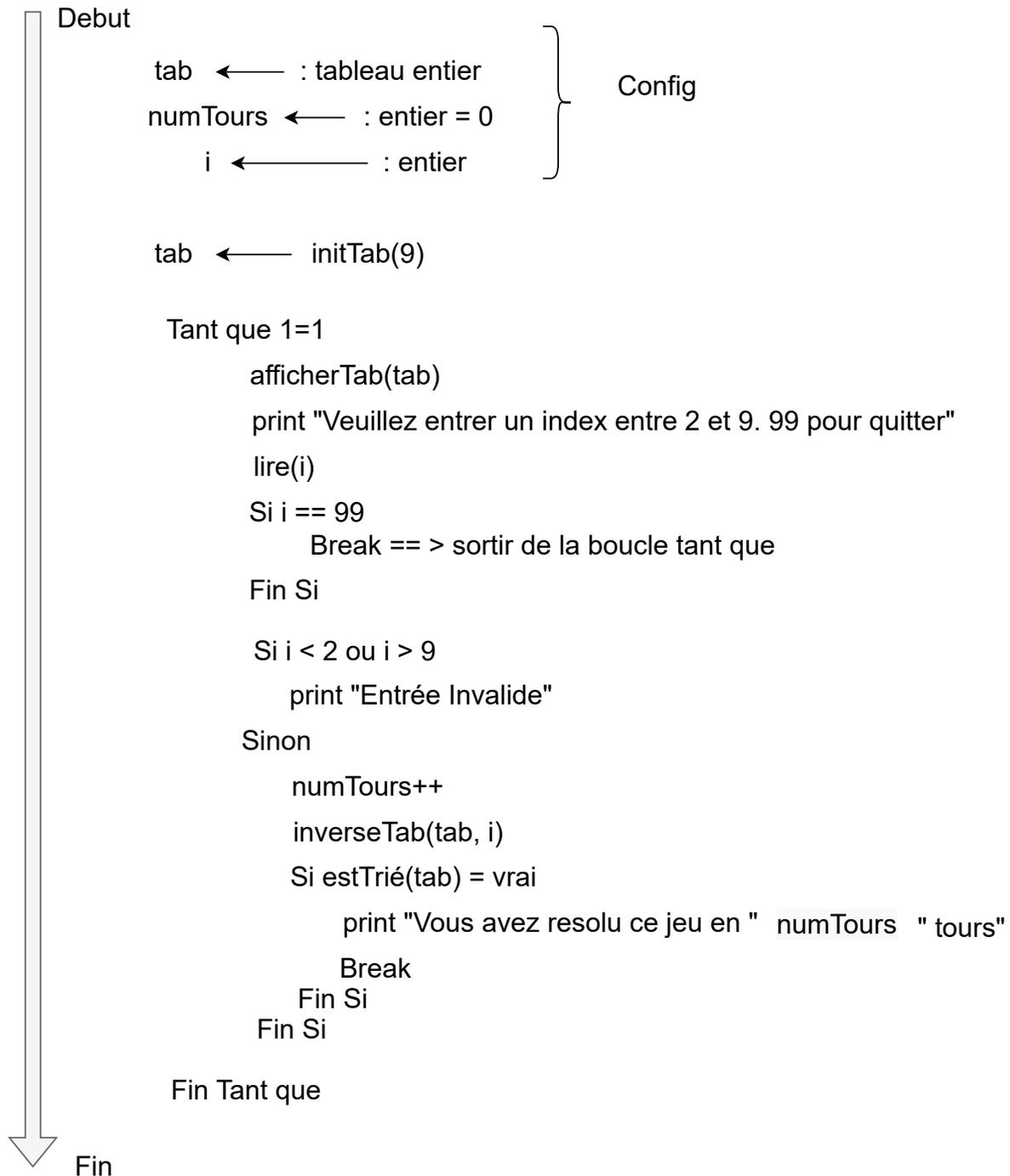
1 2 6 4 5 7 9 8 3

Le jeu s'arrête quand on atteint une configuration où les chiffres sont placés par ordre croissant. L'objectif du joueur est de réaliser ce but avec le moins de tours possible. L'algorithme permettra d'imprimer en combien de tours le joueur a réalisé le tri et lui permettra de recommencer le jeu sur la même configuration initiale autant de fois qu'il le désire. La configuration initiale est supposée fournie par une fonction:

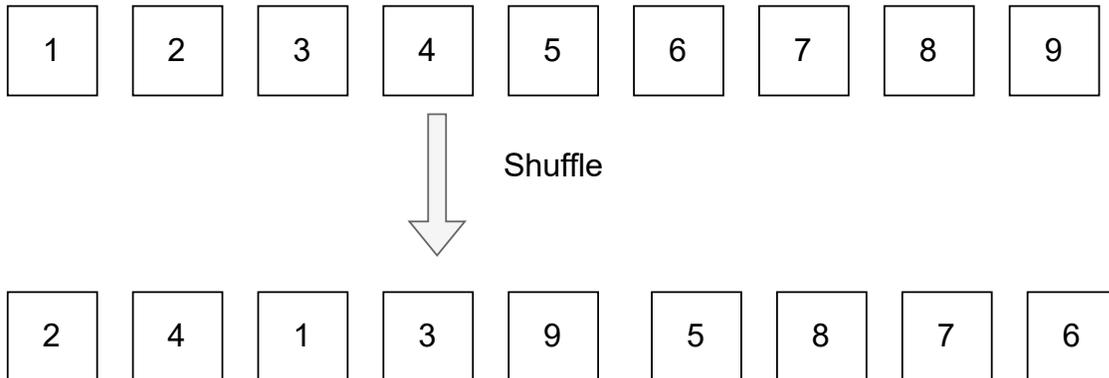
```
fonction initTab(out Tab : entier tableau, in taille : entier) : vide
```

dont on ne décrira pas l'algorithme.

Un premier pseudo-code



initTab



Algorithme

pour i allant de 0 à N-1

 tab[i] = i + 1

 melangerTab(tab)

Java

```
static int[] initTab(int N){
    int[] t = new int[N];
    for (int i = 0; i < N; i++){
        t[i] = i + 1;
    }
    melangerTab(t,N);
    return t;
}
```

melangerTab

Approche 3 :

Le tableau est parcouru de gauche à droite, et la valeur de chaque case est échangée avec celle d'une autre case se trouvant après elle dans le tableau. Cette autre case est choisie de manière aléatoire. On utilise pour cela une fonction qui retourne un nombre aléatoire.

Algorithme

Pour i allant de 0 à $N-2$

$j \leftarrow \text{nombreAleatoire}(i, N)$ $[i, N[$

$\text{echanger}(\text{tab}, i, j)$

Java

```
static void echanger(int[] tab, int i, int j){
    int tmp = tab[i];
    tab[i] = tab[j];
    tab[j] = tmp;
}
```

```
static void melangerTab(int[] tab, int N){
    for (int i = 0; i < N - 1; i++){
        int j = nombreAleatoire(i,N);
        echanger(tab,i,j);
    }
}
```

nombreAleatoire (min, max) [min, max[

Algorithme

intervalle ← (max - min) Remarque: (max-min) + 1 si [min, max]

j ← (Math.random x intervalle) + min Remarque: j entier

Exemple

min = 0

max = 2

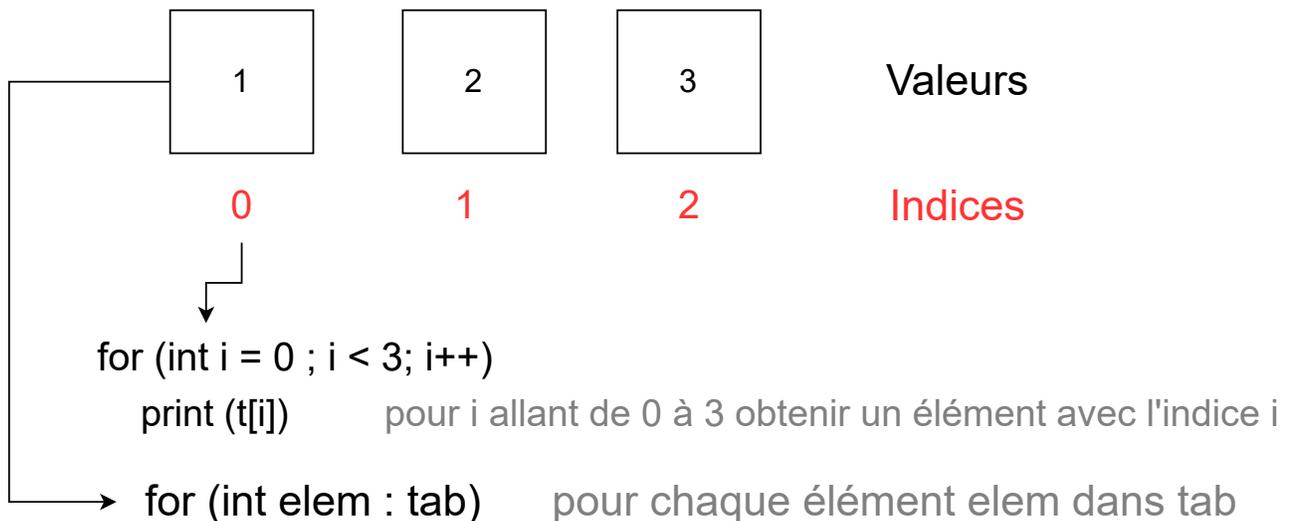
intervalle = 2 - 0 = 2

j = int(0.8 x 2) + 0 = int(1.6) + 0 = 1 ← Math.random = 0.8

j = int(0.4 x 2) + 0 = int(0.8) + 0 = 0 ← Math.random = 0.4

```
static int nombreAleatoire(int min, int max){  
    int intervalle = (max - min) ;  
    return (int)(Math.random() * intervalle) + min;  
}
```

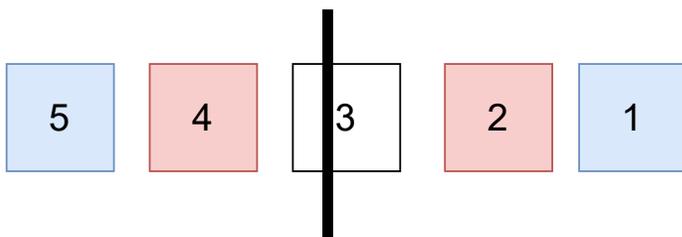
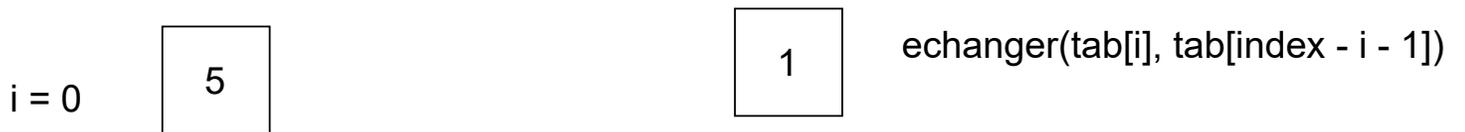
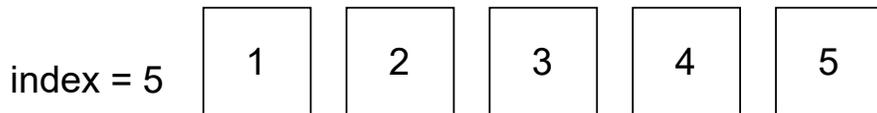
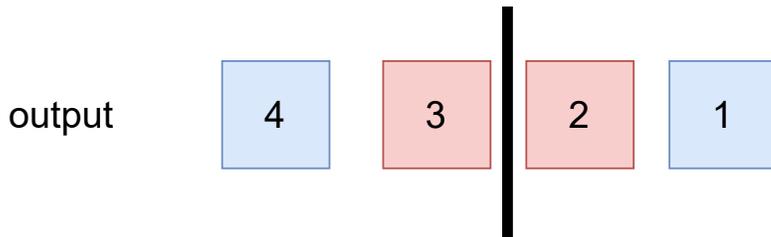
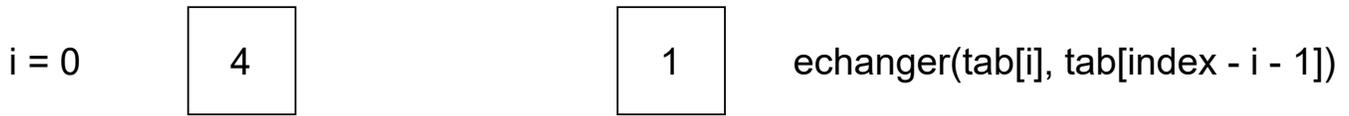
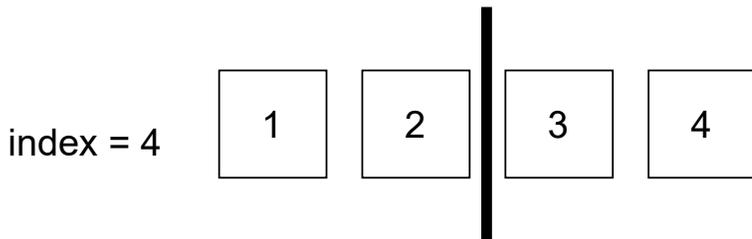
Remarque : Différence entre pour et pour chaque (for et for-each)



```
static void afficherTab(int[] tab){  
    for(int i : tab){  
        System.out.print(i);  
        System.out.print(" ");  
    }  
    System.out.println();  
}
```

Pour chaque en java

inverse (exemple)



inverse

Algorithme

index (entrée)

pour i allant de 0 à $\text{int}(\text{index}/2)$

 echanger(tab[i], tab[index - i - 1])

Java

```
static void inverse(int [] tab, int index)
{
    for (int i = 0; i < index / 2; i++)
    {
        echanger(tab,i,index-1-i);
    }
}
```

estTrie



Algorithme

Pour i allant de 0 à N - 1

Si $\text{tab}[i] > \text{tab}[i+1]$
retourner false

Fin si

retourner true

Java

```
static boolean estTrie(int[] tab){
    for (int i = 0; i < tab.length - 1; i++){
        if (tab[i] > tab[i + 1]){
            return false;
        }
    }
    return true;
}
```

Fonction main

```
public static void main(String[] args){
    int[] tab;
    int numTours = 0;
    int i = 0;
    tab = initTab(9);
    Scanner scanner = new Scanner(System.in);
    // jeu
    while(true){
        afficherTab(tab);
        System.out.println(
            "Veuillez entrer un index pour inverser"
            + " entre 2 et 9. Entrez 99 pour quitter."
        );
        i = scanner.nextInt();
        if(i == 99){
            break;
        }
        if(i < 2 || i > 9){
            System.out.println(
                "Entrée invalide!!"
            );
        } else{
            numTours++;
            inverse(tab,i);
            if (estTrie(tab)) {
                System.out.println(
                    "Félicitations, vous avez resolu ce jeu en " +
                    numTours + " tours!"
                );
                break;
            }
        }
    }
}
```

Défi : inverser un sous-tableau compte tenu de l'index et du nombre de pas vers la gauche

Exemple

index = 7

pas = 3

