

Aucun document ou support autre que le sujet ou les copies d'examen n'est autorisé.  
 (la copie ou les brouillons du voisin ne sont pas des supports autorisés).  
 Éteignez impérativement vos mobiles.

Lorsque des calculs sont nécessaires, il est impératif de les présenter sur la feuille d'examen. Il est aussi nécessaire de **justifier** ses réponses.

## 1 Exercice

On souhaite représenter en Java une structure de données stockant les éléments par lignes et colonnes, comme :

1	2	3	4
5	6	7	8

La machine utilise 4 octets pour représenter les entiers et 8 octets pour une référence.

On souhaite manipuler une telle structure de données à l'aide d'une variable de nom `tab` :

1. Quel est le type de la variable `tab` ?
2. Quelle(s) instruction(s) permettent de définir `tab` et son contenu ?
3. Dessiner à la manière du cours et des TD, le contenu de la mémoire correspondant à la variable `tab` et les données qui lui correspondent. Précisez les tailles et les zones mémoires qui correspondent aux différents objets.

## 2 Exercice

Écrire les fonctions Java suivantes :

1. `boolean estRectangulaire(int [][] t)` permettant de tester si un « tableau d'entiers » à deux dimensions et passé en paramètre est rectangulaire (toutes les lignes ont même longueur). Précisez le nombre de variables locales que votre fonction utilise.
2. `int [][] transpose(int [][] t)` qui reçoit en paramètre un « tableau à deux dimensions  $l$  et  $h$  », vérifie si celui-ci est rectangulaire et dans le cas positif renvoie en retour un autre « tableau à deux dimensions  $h$  et  $l$  » rectangulaire dont les éléments sont transposés (les lignes sont transformées en colonnes et les colonnes en lignes) et sinon dans le cas négatif renvoie une référence nulle.

Un exemple de transposition consiste à transformer le « tableau » suivant :

1	2	3	4
5	666	7	8

en le « tableau » :

1	5
2	666
3	7
4	8

## 3 Exercice

On rappelle qu'une pile Java est du type `Stack<Type_d_élément>` et possède les opérations : `Type_d_élément pop()` qui dépile et renvoie l'élément en sommet de pile (et provoque une erreur si la pile est vide), `push(Type_d_élément)` qui ajoute un élément en sommet de pile, `boolean isEmpty()` qui permet de déterminer si la pile contient des éléments ou non. La construction d'une pile vide s'effectue avec l'instruction `new Stack<Type_d_élément>()`.

1. Écrire une fonction Java `Stack<Integer> remplir(int [] t)` qui reçoit un « tableau d'entiers » en paramètre, crée une pile d'entiers et remplit la pile avec les éléments du « tableau » pris un par un en commençant par celui d'indice 0 jusqu'au dernier.
2. Dessiner le contenu de la pile renvoyée par la fonction précédente si le « tableau » contient les éléments 1, 3, 7, 11, 666, 333.
3. Écrire une fonction Java `void separe(Stack<Integer> s, Stack<Integer> p, Stack<Integer> i)` qui reçoit en paramètre une pile `s` et sépare les éléments pairs de `s` pour les placer dans `p` et les éléments impairs dans `i`, en assurant que les piles `p` et `i` ne contiennent que les éléments séparés et rien d'autre. L'ordre des éléments de `p` et `i` n'est pas important, vous pouvez choisir celui qui vous convient.

4. Dessiner l'état de la mémoire (après la dernière instruction) si l'on exécute le code suivant :

```
int [] t = { 1, 2, 3, 4, 5, 6, 7 };
Stack<Integer> p = new Stack<Integer>();
Stack<Integer> i = new Stack<Integer>();
separe(remplir(t),p,i);
```

5. Qu'afficherait l'instruction `System.out.println(i.pop());` après l'exécution du code précédent ?

## 4 Exercice

Soit le programme Java suivant :

```
public class Prog {
    public static int i;
    public static int j;
    public static void main(String []a) {
        j = 0;
        for (i=1; i<6; i++) {
            j = i*i-1;
            System.out.println(j);
        }
    }
}
```

1. Qu'affiche le programme si on l'exécute ?
2. De combien de variables ce programme a-t-il besoin pour fonctionner ?
3. Traduire, tel qu'en cours, le programme en un programme ne contenant qu'un tableau d'entiers appelé `memoire`, un entier appelé `ins`, qu'une seule boucle `while` et une seule instruction `switch` (et pas de boucle `for` évidemment!!!).
4. Qu'est ce que cela changerait à la traduction si dans le programme original on déplaçait l'instruction `System.out.println(j);` après la boucle `for` ?