

Module 1 – Informatique – Algorithmique et Programmation Objet

Travaux Pratiques (6), Licence 1ère Année

Les Tableaux

Exercice 1 Manipulation de Vecteur

Soit un vecteur (tableau à une dimension) **vect** ayant **n** valeurs positives. On désire connaître le nombre d'apparitions de chacune de ces valeurs et le mode (la fréquence la plus élevée).

- 1.1. Écrire une méthode **int [] genererVecteur(int n, int valMax)** qui accepte deux paramètres entiers **n** et **valMax**, crée un vecteur de **n** éléments, initialise chacun des éléments avec une valeur positive aléatoire comprise dans l'intervalle **[0, valMax]** et retourne le vecteur comme résultat.

Note : pour les tests prendre **n = valMax = 10**.

- 1.2. Écrire une méthode **void afficherVecteur(int [] vect)** qui accepte en entrée un vecteur d'entiers **vect** et qui affiche ses valeurs sur la console.

Exemple

```

Problems @ Javadoc Declaration Console
<terminated> TD_4_Tableaux [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (14 août 2018 14:43:18)
Contenu du vecteur :
    { 1 , 0 , 3 , 1 , 4 , 3 , 1 , 0 , 3 , 1 }

```

- 1.3. Écrire une méthode **int chercherMax(int [] vect)** qui accepte en entrée un vecteur d'entiers **vect** et qui renvoie la **valeur maximum** contenue dans le vecteur.

Exemple

```

Problems @ Javadoc Declaration Console
<terminated> TD_4_Tableaux [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (14 août 2018 14:43:18)
Contenu du vecteur :
    { 1 , 0 , 3 , 1 , 4 , 3 , 1 , 0 , 3 , 1 }
La valeur maximum du vecteur est : { 4 }

```

- 1.4. Écrire une méthode **int [] cardinalite(int [] vect)** qui renvoie le **vecteur des cardinalités** de chaque valeur contenue dans le vecteur **vect** passé en paramètre. À l'indice **i**, le vecteur des cardinalités contiendra le nombre de fois où la valeur **i** apparaît dans le vecteur **vect** passé en paramètre.

Exemple

Si **vect** =

1	0	3	1	4	3	1	0	3	1
---	---	---	---	---	---	---	---	---	---

Alors l'appel à la méthode doit retourner le vecteur

2	4	0	3	1
0	1	2	3	4

Exemple

```

Problems @ Javadoc Declaration Console
<terminated> TD_4_Tableaux [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (14 août 2018 16:37:32)
Contenu du vecteur :
    { 1 , 0 , 3 , 1 , 4 , 3 , 1 , 0 , 3 , 1 }
La valeur maximum du vecteur est : { 4 }
Contenu du vecteur cardinalités :
    { 2 , 4 , 0 , 3 , 1 }
  
```

- 1.5. Écrire une méthode **void mode(int [] vect)** qui affiche le(s) **mode(s)** (valeur(s) la/les plus fréquente(s)) du vecteur **vect** passé en paramètre.

Exemple

```

Problems @ Javadoc Declaration Console
<terminated> TD_4_Tableaux [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (14 août 2018 16:46:10)
Contenu du vecteur :
    { 1 , 0 , 3 , 1 , 4 , 3 , 1 , 0 , 3 , 1 }
La valeur maximum du vecteur est : { 4 }
Contenu du vecteur cardinalités :
    { 2 , 4 , 0 , 3 , 1 }
Le(s) mode(s) est(sont) : { 1 }
  
```

Exercice 2 Manipulation de matrices

- 2.1. Écrire une méthode **int [][] genererMatrice(int l, int c, int valMax)** qui retourne une matrice de taille **l x c**. Les cellules de la matrice générée ont des valeurs comprises entre **0** et une valeur maximum **valMax**. Les paramètres **l** et **c** sont respectivement le nombre de lignes et le nombre de colonnes de la matrice. La méthode reçoit en entrée les trois paramètres **l**, **c** et **valMax**.
- 2.2. Écrire une méthode **void afficherMatrice(int [][] mat)** qui affiche les cellules de la matrice **mat** passée en paramètre.

Note : Pour connaître le nombre de lignes et de colonnes d'une matrice **mat**, vous pouvez utiliser **mat.length** (donne le nombre de lignes) et **mat[0].length** (donne le nombre de colonnes).

Exemple

```

Problems @ Javadoc Declaration Console
<terminated> TD_4_Tableaux [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (15 août 2018 14:54:27)
Contenu de la matrice :
    { 5    10    4    20    20    8 }
    { 3    13    11   20    5    0 }
    { 5    19    3     6    15    2 }
    { 7     5     8    10    13    1 }
  
```

- 2.3. Écrire une méthode **int chercherMax(int [][] mat)** qui renvoie la valeur maximale de la matrice **mat** passée en paramètre.

Exemple

```

Problems @ Javadoc Declaration Console
<terminated> TD_4_Tableaux [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (15 août 2018 14:57:13)
Contenu de la matrice :
{ 5 10 4 20 20 8 }
{ 3 13 11 20 5 0 }
{ 5 19 3 6 15 2 }
{ 7 5 8 10 13 1 }
La valeur maximum de la matrice est : { 20 }
    
```

2.4. Écrire une méthode **int [] maxLigne(int [][] mat)** qui renvoie, pour la matrice **mat** en entrée, un vecteur contenant dans chaque cellule d'indice **i** la valeur maximum de la ligne d'indice **i** de la matrice **mat**.

Exemple

```

Problems @ Javadoc Declaration Console
<terminated> TD_4_Tableaux [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (15 août 2018 14:59:53)
Contenu de la matrice :
{ 5 10 4 20 20 8 }
{ 3 13 11 20 5 0 }
{ 5 19 3 6 15 2 }
{ 7 5 8 10 13 1 }
Contenu du vecteur rendu par maxLigne : { 20, 20, 19, 13 }
    
```

2.5. Écrire la méthode similaire **int [] maxColonne (int [][] mat)** pour les colonnes.

Exemple

```

Problems @ Javadoc Declaration Console
<terminated> TD_4_Tableaux [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (15 août 2018 15:04:19)
Contenu de la matrice :
{ 5 10 4 20 20 8 }
{ 3 13 11 20 5 0 }
{ 5 19 3 6 15 2 }
{ 7 5 8 10 13 1 }
Contenu du vecteur rendu par maxColonne : { 7, 19, 11, 20, 20, 8 }
    
```

2.6. Écrire une méthode **int [][] additionnerMat(int [][] mat1, int [][] mat2)** qui additionne deux matrices **mat1** et **mat2** de même taille passées en paramètres. La méthode renvoie une matrice correspondant au résultat de l'addition.

Exemple

```

Problems @ Javadoc Declaration Console
<terminated> TD_4_Tableaux [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (13 nov. 2018 12:14:02)
Contenus des matrices mat1 et mat2 :
{ 5 10 4 20 20 8 } { 2 9 18 19 10 88 }
{ 3 13 11 20 5 0 } { 13 3 1 15 7 10 }
{ 5 19 3 6 15 2 } { 4 9 13 16 7 12 }
{ 7 5 8 10 13 1 } { 9 4 12 9 4 16 }
← mat1 → ← mat2 →
Contenu de la matrice = mat1 + mat2 :
{ 7 19 22 30 96 }
{ 16 16 12 35 10 }
{ 9 28 16 22 14 }
{ 16 9 20 19 17 }
← mat1 + mat2 →
    
```