

```

1  /*
2  -----
3  Fichier      : matrice.h
4  Auteur(s)   : Mario Amos & Théo Coutaudier
5  Date        : 08.12.2021
6  But         : Librairie mettant à disposition des utilitaires pour le traitement
7                de vecteur ou matrice
8
9  Compilateur : Mingw-w64 g++ 11.2.0
10 -----
11 */
12
13 #ifndef INC_07_MATRICE_MATRICE_H
14 #define INC_07_MATRICE_MATRICE_H
15
16 #include <vector>
17
18 using Vecteur = std::vector<int>;
19 using Matrice = std::vector<Vecteur>;
20
21 /**
22  * Affiche un vecteur au format (x,x,x)
23  * @param os Flux de sortie
24  * @param v  Vecteur
25  * @return   Flux de sortie
26  */
27 std::ostream& operator<< (std::ostream& os, const Vecteur& v);
28
29 /**
30  * Affiche une matrice au format [(x,x,x), (x,x,x)]
31  * @param os Flux de sortie
32  * @param v  Vecteur
33  * @return   Flux de sortie
34  */
35 std::ostream& operator<< (std::ostream& os, const Matrice& m);
36
37 /**
38  * Retourne la longueur minimum des vecteurs d'une matrice
39  * @param m Matrice
40  * @return  Longueur minimum des vecteurs d'une matrice
41  */
42 unsigned minCol(const Matrice& m);
43
44 /**
45  * Retourne un vecteur contenant la somme des valeurs de chacune des lignes.
46  * @param m Matrice
47  * @return  Vecteur contenant la somme des valeurs de chacune des lignes
48  */
49 Vecteur sommeLigne(const Matrice& m);
50
51 /**
52  * Retourne un vecteur contenant la somme des valeurs de chacune des colonnes
53  * @param m Matrice
54  * @return  vecteur contenant la somme des valeurs de chacune des colonnes
55  */
56 Vecteur sommeColonne(const Matrice& m);
57
58 /**
59  * Retourne le vecteur d'une matrice dont la somme des valeurs est la plus faible.
60  * @param m Matrice
61  * @return  Vecteur d'une matrice dont la somme des valeurs est la plus faible
62  */
63 Vecteur vectSommeMin(const Matrice& m);
64
65 /**
66  * Vérifie si tout les vecteurs d'une matrice ont la même taille.
67  * @param m Matrice
68  * @return  Un bool indiquant si tout les vecteurs ont la même taille
69  */
70 bool estReguliere(const Matrice& m);
71
72 /**

```

```
73  * Vérifie si la matrice estCarré, si la matrice est vide elle est considéré carré.
74  * @param m Matrice
75  * @return un bool indiquant si la matrice est carré ou non.
76  */
77  bool estCarre(const Matrice& m);
78
79  /**
80   * Melange les vecteurs d'une matrice en fonction d'un clock, ne change pas
81   * l'ordre à l'interieur des vecteurs.
82   * @param m Matrice
83   * @return
84   */
85  void shuffleMatrice(Matrice& m);
86
87  /**
88   * Trie la matrice en fonction du min_element de chaque vecteur
89   * @param m Matrice
90   * @return
91   */
92  void sortMatrice(Matrice& m);
93
94  #endif //INC_07_MATRICE_MATRICE_H
95
```