

/*

Nom du fichier : *affichageTableau.cpp*
Nom du Labo : *HEIGVD-PRG1-CRIBLE*
Auteur(s) : *Jérémie Métrailler, Nikola Bouattit*
Date creation : *19.11.2021*

Description : *Les fonctions permettant l'affichage de la matrice basé sur un tableau 1 dimension*

Remarque(s) : *Seulement les tableaux a une seule dimension sont supporté
 Les matriche affiche pour chaque ligne le nombre maximal de
 colonne*

Assertions : *n/a*

Modifications : *n/a*

Compilateur : *Mingw-w64 g++ 11.2.0*

*/

```
#include "affichageTableau.h"
#include <iostream>      //Nécessaire pour l'écriture dans la console
#include <iomanip>        //Nécessaire pour limiter l'affichage

const char REPRESENTANT = '0'; // Caractere pour l'affichage de la matrice
const int ESPACE_AFFICHAGE = 3; // Espace définie pour un affichage constant
```

```
using namespace std;
```

```
void afficherTableau(const int tab[], unsigned taille, unsigned nbColonne,
                    char charSpecial){
```

```
    unsigned caseTableau = 0;
    if(nbColonne == 0)
    {
        for (unsigned ligne = 1; ligne <= taille; ++ligne) {
            if (ligne == (unsigned)tab[caseTableau]) {
                cout << fixed << left
                    << setw(ESPACE_AFFICHAGE) << REPRESENTANT;
                ++caseTableau;
            } else {
                cout << fixed << left
                    << setw(ESPACE_AFFICHAGE) << charSpecial;
            }
        }
        cout << endl;
    }
    else {
        for (unsigned ligne = 1; ligne <= taille;) {
            for (unsigned colonne = 0; colonne < nbColonne; ++colonne) {
                if (ligne == (unsigned)tab[caseTableau]) {
                    cout << fixed << left
                        << setw(ESPACE_AFFICHAGE) << REPRESENTANT;
                    ++caseTableau;
                } else {
                    cout << fixed << left
                        << setw(ESPACE_AFFICHAGE) << charSpecial;
                }
                ++ligne;
            }
            cout << endl;
        }
    }
}
```

```
void afficherTableau( unsigned taille, unsigned nbColonne){
    if(nbColonne == 0){
        for(unsigned i =0; i < taille;++i){
            cout << fixed << left
                << setw(ESPACE_AFFICHAGE) << REPRESENTANT;
        }
    }
    else {
        for (unsigned ligne = 0; ligne < taille; ++ligne) {
            for (unsigned colonne = 0; colonne < nbColonne; ++colonne) {
                cout << fixed << left
                    << setw(ESPACE_AFFICHAGE) << REPRESENTANT;
                ++ligne;
            }
            cout << endl;
        }
    }
}

void ecrireTableau(unsigned taille, const int tab[]) {
    for (unsigned int i = 0; i < taille; ++i) {
        cout << fixed << left << setw(ESPACE_AFFICHAGE) << tab[i];
    }
}
```