

Titre :

## Algorithme Base de la programmation

**Fait par :** Melliti Ramla

**Date :** 27 /01/2020

## **Sommaire :**

1. Rappel de la problématique.	Page 3
2. Partie A : réponse et algorithme en pseudo-code	Page 3-4
3. Partie B : Implimentation en langage Java ou C (code commenté)	Page 4-7
4. Jeux de tests.	Page 8
5. Conclusion.	Page 9

### Problème :

Une image en nuances de gris, découpée en pixels, peut être considérée comme un tableau à deux dimensions, ou une matrice, dont les éléments sont des entiers de 0 à 100, appelés **saturation**.

0 correspond à une case noire, 100 à une case blanche, un entier entre 0 et 100 à une case grise d'autant plus foncée que le nombre est proche de 0.

**L'objectif est d'écrire quelques algorithmes modifiant la saturation d'une image.**

### Partie A : réponse et algorithme en pseudo-code

#### Question 1 :

Le fonctionnement de la fonction MYSTERE() est d'inverser les nuances et la saturation . Et dans cette exemple la fonction donne l'image inversé de la Matrice M. le noir sera blanc, et le blanc sera noir ...

#### Question 2 :

**FONCTION accentuerContrast**(tableau de Entier : Matrice[[]]) :tableau de Entier

##### **VARIABLES**

Entier : i, j

##### **DEBUT**

Pour i allant de 0 à 4

Pour j allant de 0 à 5

Si ((Matrice[i][j]>50)&&(Matrice[i][j]<=75)) alors

Matrice[i][j] ← 75

Sinon-Si (Matrice[i][j]>75) alors

Matrice[i][j] ← 100

Sinon

Matrice[i][j] ← Matrice[i][j]/2

Fin\_Si

Fin\_Si

Fin\_Pour

Fin\_Pour

RETOURNE Matrice

**FIN**

## **Partie B : Implémentation en langage Java :**

```
package image;
```

```
public class Image {
```

```
    public static void main(String[] args) {
```

```
        int i,j;
```

```
        int Matrice[][]=new int [][]{
```

```
            {25,28,100,100,100,100},
```

```
            {44,45,50,100,100,100},
```

```
            {65,65,50,20,100,100},
```

```
            {90,62,100,5,0,0},
```

```
            {60,87,100,0,100,100}  };
```

```
        for(i=0;i<5;i++){
```

```
            System.out.println();
```

```
            for(j=0;j<6;j++){
```

```
                System.out.print(Matrice[i][j]+"\\t");
```

```
            }
```

```
        }
```

```
        ///*****Affichage de nuance inversé *****///
```

```
        System.out.print("\\n Nuance inversé :");
```

```
        int [][] matInv=Mystere(Matrice);
```

```
        for(int k=0;k<5;k++){
```

```

        System.out.print("\n");
        for(int l=0;l<6;l++){
            System.out.print(matInv[k][l]+"\\t");
        }
    }

    //////////***** Affichage
    moyenne*****

    System.out.print("\n La saturation moyenne vaut : "+Moyenne(Matrice));

    ///*****Affichage de matrice avec les Contrastes diminués *****

    System.out.print("\n Contrast Diminué :");
    int [][] matDim=DiminuantContrast(Matrice);
    for(i=0;i<5;i++){
        System.out.println();
        for(j=0;j<6;j++){
            System.out.print(matDim[i][j]+"\\t");
        }
    }
}

```

```

////////***** fonction Mystere *****

public static int[][] Mystere(int [][]M){

```

```

    //int [][] matrice1=new int[5][6];

    int matrice1[][]=new int [][]{
        {25,28,100,100,100,100},
        {44,45,50,100,100,100},
        {65,65,50,20,100,100},

```

```

        {90,62,100,5,0,0},
        {60,87,100,0,100,100}  };

for(int i=0;i<5;i++){
    for(int j=0;j<6;j++){
        matrice1[i][j]=100-matrice1[i][j];
    }
}

return matrice1;
}

```

////////\*\*\*\*\* fonction Moyenne \*\*\*\*\*

```

public static double Moyenne(int[][]M){
    int matrice1[][]=new int [][]{
        {25,28,100,100,100,100},
        {44,45,50,100,100,100},
        {65,65,50,20,100,100},
        {90,62,100,5,0,0},
        {60,87,100,0,100,100}  };

    int som=0;
    double moy;
    for(int i=0;i<5;i++){
        for(int j=0;j<6;j++){
            som=som+matrice1[i][j];
        }
    }

    moy=som/30.0;
    return moy;
}

```

//////////\*\*\*\*\*Fonction Diminuant Contrast\*\*\*\*\*

```
public static int[][] DiminuantContrast(int [][]M){

    int matrice2[][]=new int [][]{

        {25,28,100,100,100,100},
        {44,45,50,100,100,100},
        {65,65,50,20,100,100},
        {90,62,100,5,0,0},
        {60,87,100,0,100,100}  };

    for(int i=0;i<5;i++){
        for(int j=0;j<6;j++){
            if(matrice2[i][j]>Moyenne(matrice2))
                matrice2[i][j]=50;
            else
                matrice2[i][j]=matrice2[i][j]/2;
        }
    }
    return matrice2;
}
```

## Jeux de test :

### Essai avec n°1

```
Output - Image (run) x
run:
25      28      100      100      100      100
44      45      50      100      100      100
65      65      50      20      100      100
90      62      100      5      0      0
60      87      100      0      100      100
  Nuance inversé :
75      72      0      0      0      0
56      55      50      0      0      0
35      35      50      80      0      0
10      38      0      95      100      100
40      13      0      100      0      0
  La saturation moyenne vaut : 66.53333333333333
  Contrast Diminué :
12      14      50      50      50      50
22      22      25      50      50      50
32      32      25      10      50      50
50      31      50      2      0      0
30      50      50      0      50      50      BUILD SUCCESSFUL (total time: 0 seconds)
```

## Conclusion:

Une image en nuances de gris, découpée en pixels, peut être considérée comme un tableau à deux dimensions, ou une matrice, dont les éléments sont des entiers de 0 à 100, appelés **saturation**.

**On peut changer les contrastes et la saturation avec cet algorithme.**