

# TPQ1

---

## Exercice 1

Correction des erreurs :

1)

```
boolean toutPair = FAUX;
```

l'erreur était un erreur de signalisation la variable toutPair était initialiser a 0 et non a un boolean

2)

```
public final static double seuil = 1.76;
```

erreur d'initialisation sa variable seuil était considéré comme un float est non un double

3)

```
toutPair = true;
```

erreur de syntaxe : la variable toutPair s'est retrouver initialiser a la valeur **trou** au lieu de **true**

4)

```
if (laveurs[cpt] % 2==0 )
```

l'erreur ici se trouve être une erreur de syntaxe ou le programmeur avait oublié le ==0 à la fin par conséquent le programme testait si "laveurs[cpt] % 2" renvoyait une valeur boolean true ou false

5)

```
static boolean toutPair = false;
```

l'erreur ici est une erreur de static la variable toutPair s'est trouver ne pas être static ce qui a entraîné des erreurs

6)

```
static double moyenne ()
```

erreur static du même type que si dessus

7)

```
static int[] laveurs = {2, 0, 0, 2};
```

erreur de type contexte statique la variable laveurs n'était pas considéré comme static

8)

```
public static double seuil = 1.76;
```

la variable était considéré comme étant une variable final alors qu'elle ne devait pas s'être donc une erreur d'initialisation

9)

```
int i =0 ;
```

l'erreur ici est une erreur d'initialisation la variable i n'était tout simplement pas initialiser a 0

10)

```
Decouverte autreObjet= new Decouverte ();
```

l'erreur ici se trouve aussi être une erreur d'initialisation l'objet autreObjet n'était pas considéré comme étant directement un nouveau objet de découverte

11)

```
while (i < laveurs.length)
```

l'erreur ici s'est trouvée être que l'index allait jusqu'à 4 ce qui sortait du tableau car il considérait qu'il y avait donc 5 éléments pour la solution nous avons utilisé la longueur du tableau pour éviter de sortir du tableau

12)

```
double m = monObjet.moyenne();
```

définition d'un objet pour éviter les autres erreurs.

Maintenant que le programme fonctionne nous allons nous attarder sur les mots `Infinity` et `NaN`

`Infinity` apparaît car l'on initialise la variable `autreObjet` à une division par 0

quant à `NaN` il apparaît car l'on trouve une erreur lors de l'opération suivante : `"autreObjet.seuil - autreObjet.seuil"`

## Exercice 2

Le programme `terminator.java` permet de mettre en évidence les limites du ramasse-miettes. Le nom contrôle du ramasse-miettes est mis en évidence à l'aide de plusieurs problèmes. Par exemple si l'on rentre plusieurs fois le nombre 1 le ramasse-miettes se lance automatiquement de plus l'autre dans le quelle il détruit les objets et aléatoirement, de plus il peut effectuer la destruction n'importe quand.