

Rapport SAE 2.02



BLAGNAC

IBRAHIM Marwane

Efficacité :

11efficacite.java : 18

Issues 2

Filter All categories All levels All patterns

MINOR	Code Style	The static method name 'RLE' doesn't match '[a-z][a-zA-Z0-9]*'	▼
4	<code>public static String RLE(String in) {</code>		
MINOR	Code Style	The static method name 'RLE' doesn't match '[a-z][a-zA-Z0-9]*'	▼
32	<code>public static String RLE(String in, int iteration) throws AlgoException{</code>		

Tous les tests initiaux sont passés et il n'y a aucun problème majeur si ce n'est le nom des fonctions comme montré par codacy qui ne respectent pas les conventions.

13efficacite.c : 18

Issues 0 Duplication 4

Your file does not have any issues.

Tous les tests passent et la complexité est de $O(n)^2$.

32efficacite.java : 18

Pas de problème

47efficacite.java : 16

Issues 5

Filter All categories All levels All patterns

MINOR	Code Style	The utility class name 'efficacite' doesn't match '[A-Z][a-zA-Z0-9]*'	▼
3	<code>public class efficacite {</code>		
MINOR	Code Style	The static method name 'RLE' doesn't match '[a-z][a-zA-Z0-9]*'	▼
6	<code>public static String RLE(String entree) {</code>		
MINOR	Code Style	The static method name 'RLE' doesn't match '[a-z][a-zA-Z0-9]*'	▼
41	<code>public static String RLE(String entree, int iteration) {</code>		
MEDIUM	Code Style	Avoid reassigning parameters such as 'entree'	▼
47	<code>entree = RLE(entree); // Applique la compression RLE plusieurs fois</code>		
MEDIUM	Code Style	Avoid reassigning parameters such as 'entree'	▼
82	<code>entree = unRLE(entree); // Applique la décompression RLE plusieurs fois</code>		

Les tests passent mais il y a quelques problèmes. L'attribut "entree" n'a pas besoin d'être réassigné(Codacy).

65efficacite.java : 10

```
|| MEDIUM | Code Style | Avoid reassigning parameters such as 'i' |
31 while (i-- > 0) {
```

Petite issue soulignée par codacy, il est déconseillé de réassigner la valeur dans le while. De plus, le code ne passe pas les tests fournis initialement.

Simplicité:

12simplicite.java : 18

Pas d'issue pour Codacy.

Tous les tests passent pour les RLE et unRLE le code est facile à lire et les noms de variables sont très claires.

20simplicite.py : 10

Pas d'issue importante par Codacy.

Ne passe pas les tests initiaux.

21simplicite.java : 18

```
Issues 3
Filter  All categories  All levels  All patterns
|| MINOR | Code Style | The static method name 'RLE' doesn't match '[a-z][a-zA-Z0-9]*' |
5 public static String RLE(String in) {

|| MINOR | Code Style | The static method name 'RLE' doesn't match '[a-z][a-zA-Z0-9]*' |
57 public static String RLE(String in, int iteration) throws AlgoException {

|| MEDIUM | Performance | Avoid calling toString() on String objects; this is unnecessary. |
98 return result.toString();
```

Passe les tests initialement fournis, mais appelle toString sur un String, ce qui fait perdre un peu de performance.

32simplicite.py : 10

Issues 23





Echoue sur le test “SAE Algo” pour RLE_Recursif, de plus il y a des commentaires à chaque ligne de code entre les lignes ce qui rend le code compliqué à lire.

66simplicite.java : 18

Issues 2

Filter  All categories  All levels  All patterns 



	MINOR	Code Style	The static method name 'RLE' doesn't match '[a-z][a-zA-Z0-9]*'	
5	<code>public static String RLE(String input) {</code>			
	MINOR	Code Style	The static method name 'RLE' doesn't match '[a-z][a-zA-Z0-9]*'	
26	<code>public static String RLE(String input, int iteration) throws AlgoException{</code>			

Passe les tests fournis initialement. Codacy relève des issue mineurs pas importante qui sont les conventions de nommage des methodes.

Sobriété:

13sobriete.c : 18

Issues 0 Duplication 2

Your file does not have any issues.











Tous les tests passent et Codacy ne relève aucune issue.

49sobriete.java : 18

Issues 4

Filter  All categories  All levels  All patterns 



	MINOR	Code Style	The utility class name 'sobriete' doesn't match '[A-Z][a-zA-Z0-9]*'	
1	<code>public class sobriete {</code>			
	MINOR	Code Style	All classes, interfaces, enums and annotations must belong to a named package	
1	<code>public class sobriete {</code>			
	MINOR	Code Style	The static method name 'RLE' doesn't match '[a-z][a-zA-Z0-9]*'	
3	<code>public static String RLE(String texte) {</code>			
	MINOR	Code Style	The static method name 'RLE' doesn't match '[a-z][a-zA-Z0-9]*'	
36	<code>public static String RLE(String chaine, int nombre) {</code>			

Tous les tests initiaux passent et Codacy ne relève pas d'erreur si ce n'est des conventions.

61sobriete.java : 18

Issues 4

Filter All categories All levels All patterns

MINOR	Code Style	The utility class name 'sobriete' doesn't match '[A-Z][a-zA-Z0-9]*'	▼
1 public class sobriete {			
MINOR	Code Style	All classes, interfaces, enums and annotations must belong to a named package	▼
1 public class sobriete {			
MINOR	Code Style	The static method name 'RLE' doesn't match '[a-z][a-zA-Z0-9]*'	▼
3 public static String RLE(String texte) {			
MINOR	Code Style	The static method name 'RLE' doesn't match '[a-z][a-zA-Z0-9]*'	▼
36 public static String RLE(String chaine, int nombre) {			

Tous les tests initiaux passent et Codacy ne relève pas d'erreur si ce n'est des conventions.

63sobriete.java : 17

Issues 3

Filter All categories All levels All patterns

MINOR	Code Style	The static method name 'RLE' doesn't match '[a-z][a-zA-Z0-9]*'	▼
5 public static String RLE(String in) {			
MINOR	Code Style	The static method name 'RLE' doesn't match '[a-z][a-zA-Z0-9]*'	▼
26 public static String RLE(String in, int iteration) throws AlgoException{			
MEDIUM	Code Style	Avoid reassigning parameters such as 'in'	▼
30 in=resultat;			

Tous les tests passent mais Codacy relève une erreur qui pour la catégorie sobriété numérique est importante.

Classement:

Efficacité :

1. 11efficacite.java
2. 32efficacite.java
3. 13efficacite.java
4. 47efficacite.c
5. 65.efficacite.java

Simplicité :

1. 12simplicite.java
2. 21simplicite.java
3. 66simplicite.java
4. 20simplicite.py
5. 32simplicite.py

Sobriété numérique :

1. 13sobriete.c
2. 49sobriete.java
3. 61sobriete.java
4. 63sobriete.java