

CARLIER FLAVIE

SAÉ2023-2-02

# LIVRABLE PHASE 2

# SOMMAIRE :

<b>Introduction</b>	<b>3</b>
<b>1 - Méthode d'évaluation choisie</b>	<b>3</b>
1.1 - Méthode d'évaluation : Efficacité	3
1.2 - Méthode d'évaluation : Simplicité	3
1.3 - Méthode d'évaluation : Sobriété	3
<b>2 - Evaluation individuelle des algorithmes</b>	<b>4</b>
2.1 - Efficacité	4
2.1.1 - Efficacité meilleur	4
2.1.1.1 - Algorithme n°24	4
2.1.1.2 - Algorithme n°62	4
2.1.2 - Efficacité pire	5
2.1.2.1 Algorithme n°34	5
2.1.2.2 Algorithme n°48	5
2.2 - Simplicité	6
2.2.1 - Simplicité meilleur	6
2.2.1.1 Algorithme n°15	6
2.2.1.2 Algorithme n°20	6
2.2.1.3 Algorithme n°40	6
2.2.2 - Simplicité pire	7
2.2.2.1 Algorithme n°44	7
2.2.2.2 Algorithme n°63-1	7
2.3 - Sobriété	8
2.3.1 - Sobriété meilleur	8
2.3.1.1 Algorithme n°17	8
2.3.1.2 Algorithme n°63	8
2.3.2 - Sobriété pire	9
2.3.2.1 Algorithme n°10	9
2.3.2.2 Algorithme n°48	9
<b>3 - Comparaison des algorithmes</b>	<b>10</b>
3.1 - Efficacité	10
3.1.1 - Efficacité meilleur	10

3.1.1.1 - Algorithme n°24	3.1.1.2 - Algorithme n°62	10
3.1.2 - Efficacité pire		11
3.1.2.1 Algorithme n°34.	3.1.2.2 Algorithme n°48	11
3.2 - Simplicité		12
3.2.1 - Simplicité meilleur		12
3.2.1.1 Algorithme n°15	3.2.1.2 Algorithme n°20	12
2.2.1.3 Algorithme n°40		12
3.2.2 - Simplicité pire		13
3.2.2.1 Algorithme n°44	3.2.2.2 Algorithme n°63-1	13
3.3 - Sobriété		14
3.3.1 - Sobriété meilleur		14
3.3.1.1 Algorithme n°17	3.3.1.2 Algorithme n°63	14
3.3.2 - Sobriété pire		15
3.3.2.1 Algorithme n°10	3.3.2.2 Algorithme n°48	15
<b>4 - Classement des algorithmes</b>		<b>16</b>
Classement efficacité		16
Classement simplicité		16
Classement sobriété		16
<b>Conclusion</b>		<b>17</b>

# INTRODUCTION

La phase 2 de cette SAÉ (Situation d'Apprentissage et d'Évaluation) vise à approfondir la réflexion sur l'approche algorithmique des problèmes rencontrés pendant les phases de développement. Après avoir soumis nos solutions individuelles pour résoudre le problème de classement des mots dans un texte, nous entrons maintenant dans une étape cruciale : la comparaison et l'évaluation des différentes solutions proposées.

L'objectif de cette deuxième phase est de mettre en place un processus de classement et d'analyse des algorithmes, en binôme, selon différents critères préalablement définis. Nous devons maintenant évaluer chaque solution sur la base de la lisibilité du code, de la qualité du code, de l'efficacité, de la sobriété numérique et du temps d'exécution.

## 1 - MÉTHODE D'ÉVALUATION CHOISIE

### 1.1 - MÉTHODE D'ÉVALUATION : EFFICACITÉ

Dans notre évaluation de l'efficacité des solutions algorithmiques, nous avons opté pour deux critères principaux : la rapidité d'exécution et la note Codacy attribuée à chaque algorithme. Ces deux aspects nous permettront d'évaluer à la fois les performances et la qualité du code des différentes solutions proposées.

### 1.2 - MÉTHODE D'ÉVALUATION : SIMPLICITÉ

Dans notre évaluation de la simplicité des solutions algorithmiques, nous nous concentrons sur trois aspects clés : la taille du code, sa lisibilité et sa structure. Ces critères nous permettent d'évaluer à quel point une solution est simple, facile à comprendre et à maintenir.

### 1.3 - MÉTHODE D'ÉVALUATION : SOBRIÉTÉ

Dans notre évaluation de la sobriété des solutions algorithmiques, nous mettons l'accent sur trois aspects essentiels : le respect des bonnes pratiques, la lisibilité du code et sa capacité à être réutilisé. Ces critères nous permettent d'évaluer si une solution est sobre, c'est-à-dire si elle utilise de manière optimale les ressources disponibles et facilite la maintenance et la réutilisation du code.


## 2 - EVALUATION INDIVIDUELLE DES ALGORITHMES

Dans le cadre de cette deuxième phase de notre projet, nous allons procéder à l'évaluation individuelle de chaque algorithme soumis lors de la première phase. Cette approche nous permettra d'examiner en détail chaque solution, en évaluant chaque algorithme individuellement, nous serons en mesure de comprendre leurs forces et leurs faiblesses spécifiques, ainsi que leur adéquation aux différents critères d'évaluation que nous avons définis.

### 2.1 - EFFICACITÉ

#### 2.1.1 - EFFICACITÉ MEILLEUR

##### 2.1.1.1 - ALGORITHME N°24


*Efficacité meilleur*  
N°24

☐ Respect nomenclature  
☐ Respect anonymat  
☐ Compile  
☐ Classe Exercice

Test  
☐ Erreur  
☐ Test initial  
☐ Test supplémentaires  
☐ Hors concours

**NOTE : 20/20**

##### 2.1.1.2 - ALGORITHME N°62

*Efficacité meilleur*  
N°62


☐ Respect nomenclature  
☐ Respect anonymat  
☐ Compile  
☐ Classe Exercice

Test  
☐ Erreur (->AssertionError)  
☐ Test initial  
☐ Test supplémentaires  
☐ Hors concours

**NOTE : 5/20**

## 2.1.2 - EFFICACITÉ PIRE

### 2.1.2.1 ALGORITHME N°34

*Efficacité pire*n°34

☒ Respect nomenclature

☒ Respect anonymat

☒ Compile

☒ Classe Exercice

Test

☒ Erreur


☐ Test initial

☐ Test supplémentaires

☐ Hors concours

**NOTE : 5/20**

### 2.1.2.2 ALGORITHME N°48

*Efficacité pire*n°48

☐ Respect nomenclature

☐ Respect anonymat

☐ Compile

☐ Classe Exercice

Test

☐ Erreur

☐ Test initial

☐ Test supplémentaires


☐ Hors concours

**NOTE : 4/20**

## 2.2 - SIMPLICITÉ

### 2.2.1 - SIMPLICITÉ MEILLEUR

#### 2.2.1.1 ALGORITHME N°15


*Simplicité meilleur*  
n°15

☐ Respect nomenclature  
☐ Respect anonymat  
☐ Compile  
☐ Classe Exercice

Test  
☐ Erreur  
☐ Test initial  
☐ Test supplémentaires  
☐ Hors concours

NOTE : **20/20**

#### 2.2.1.2 ALGORITHME N°20


*Simplicité meilleur*  
n°20

☐ Respect nomenclature  
☐ Respect anonymat  
☐ Compile  
☐ Classe Exercice

Test  
☐ Erreur  
☐ Test initial  
☐ Test supplémentaires  
☐ Hors concours

NOTE : **5/20**

#### 2.2.1.3 ALGORITHME N°40

*Simplicité meilleur*  
n°40


☐ Respect nomenclature  
☐ Respect anonymat  
☐ Compile  
☐ Classe Exercice

Test  
☐ Erreur  
☐ Test initial  
☐ Test supplémentaires  
☐ Hors concours

NOTE : **5/20**

## 2.2.2 - SIMPLICITÉ PIRE

### 2.2.2.1 ALGORITHME N°44



*Simplicité pire*

*n°44*


- ☒ Respect nomenclature
- ☒ Respect anonymat
- ☐ Compile (manque package)
- ☒ Classe Exercice

Test

- ☒ Erreur
- ☐ Test initial
- ☐ Test supplémentaires
- ☐ Hors concours

**NOTE : 4/20**

### 2.2.2.2 ALGORITHME N°63-1



*Simplicité pire*

*n°63-1*

- ☐ Respect nomenclature
- ☒ Respect anonymat
- ☒ Compile
- ☒ Classe Exercice

Test

- ☒ Erreur
- ☐ Test initial
- ☐ Test supplémentaires
- ☐ Hors concours


**NOTE : 4/20**



## 2.3 - SOBRIÉTÉ

### 2.3.1 - SOBRIÉTÉ MEILLEUR

#### 2.3.1.1 ALGORITHME N°17


*Sobriété meilleur*  
*N°17*

☐ Respect nomenclature  
☐ Respect anonymat  
☐ Compile  
☐ Classe Exercice

Test  
☐ Erreur  
☐ Test initial  
☐ Test supplémentaires  
☐ Hors concours

**NOTE : 5/20**

#### 2.3.1.2 ALGORITHME N°63

*Sobriété meilleur*  
*N°63*


☐ Respect nomenclature  
☐ Respect anonymat  
☐ Compile  
☐ Classe Exercice

Test  
☐ Erreur  
☐ Test initial  
☐ Test supplémentaires  
☐ Hors concours

**NOTE : 10/20**

## 2.3.2 - SOBRIÉTÉ PIRE

### 2.3.2.1 ALGORITHME N°10



*Sobriété pire*

*n°10*

☐ Respect nomenclature  
☐ Respect anonymat  
☐ Compile  
☐ Classe Exercice

Test

☐ Erreur  
☐ Test initial  
☐ Test supplémentaires  
☐ Hors concours

**NOTE : 5/20**

### 2.3.2.2 ALGORITHME N°48



*Sobriété pire*

*n°48*

☐ Respect nomenclature  
☐ Respect anonymat  
☐ Compile  
☐ Classe Exercice

Test

☐ Erreur  
☐ Test initial  
☐ Test supplémentaires  
☐ Hors concours

**NOTE : 4/20**


## 3 - COMPARAISON DES ALGORITHMES

Nous abordons la partie cruciale de la comparaison et de l'évaluation des solutions algorithmiques soumises. L'objectif est d'attribuer des notes en fonction de leur performance et de leur adéquation aux critères d'évaluation définis. Cette étape nous permettra de déterminer par la suite un classement

### 3.1 - EFFICACITÉ

#### 3.1.1 - EFFICACITÉ MEILLEUR

##### 3.1.1.1 - ALGORITHME N°24

 **N°24**

**Efficacité meilleur**

Test

0 1 3 5

Note Codacy

☐ ☐ ☒ ☐

Temps d'exécution < 50 ms

☒ ☐ ☐ ☐

Temps d'exécution < 100 ms

☒ ☐ ☐ ☐

Temps d'exécution < 200 ms


☐ ☐ ☐ ☒

Temps d'exécution < 300 ms

☐ ☐ ☐ ☒

**NOTE 13**

##### 3.1.1.2 - ALGORITHME N°62

 **N°62**

**Efficacité meilleur**

Test

0 1 3 5

Note Codacy

☐ ☐ ☒ ☐

Temps d'exécution < 100 ms

☒ ☐ ☐ ☐

Temps d'exécution < 200 ms

☒ ☐ ☐ ☐

Temps d'exécution < 300 ms

☒ ☐ ☐ ☐


Temps d'exécution < 400 ms

☐ ☐ ☐ ☒

**NOTE 8**

### 3.1.2 - EFFICACITÉ PIRE

#### 3.1.2.1 ALGORITHME N°34.

 **n°34**

**Efficacité pire**

Test

0135

Note Codacy

☐ ☐ ☒ ☐

Temps d'exécution < 100 ms

☐ ☐ ☐ ☒

Temps d'exécution < 200 ms

☐ ☐ ☐ ☒

Temps d'exécution < 300 ms


☐ ☐ ☐ ☒

Temps d'exécution < 400 ms

☐ ☐ ☐ ☒

**NOTE HC**

#### 3.1.2.2 ALGORITHME N°48

 **n°48**

**Efficacité pire**

Test

0135

Note Codacy

☐ ☐ ☒ ☐

Temps d'exécution < 100 ms

☐ ☐ ☐ ☒

Temps d'exécution < 200 ms

☐ ☐ ☐ ☒

Temps d'exécution < 300 ms

☐ ☐ ☐ ☒

Temps d'exécution < 400 ms


☐ ☐ ☐ ☒

**NOTE 23**

## 3.2 - SIMPLICITÉ

### 3.2.1 - SIMPLICITÉ MEILLEUR

#### 3.2.1.1 ALGORITHME N°15

 *N°15*

Simplicité meilleur

Test

0135

Taille courte

☐ ☒ ☐ ☐

Lisibilité des variables


☐ ☐ ☐ ☒

Structure compréhensible

☐ ☐ ☐ ☒

**NOTE 11**

#### 3.2.1.2 ALGORITHME N°20

 *N°20*

Simplicité meilleur

Test

0135

Taille courte

☐ ☐ ☒ ☐

Lisibilité des variables


☐ ☒ ☐ ☐

Structure compréhensible

☐ ☐ ☒ ☐

**NOTE 7**

#### 2.2.1.3 ALGORITHME N°40

 *N°40*

Simplicité meilleur

Test

0135

Taille courte

☐ ☐ ☒ ☐

Lisibilité des variables

☐ ☐ ☐ ☒


Structure compréhensible

☐ ☐ ☐ ☒

**NOTE 13**

## 3.2.2 - SIMPLICITÉ PIRE

### 3.2.2.1 ALGORITHME N°44

 **N°44**

**Simplicité pire**

Test

0135

Taille courte

☐ ☐ ☐ ☒

Lisibilité des variables


☐ ☒ ☐ ☐

Structure compréhensible

☒ ☐ ☐ ☐

**NOTE 6**

### 3.2.2.2 ALGORITHME N°63-1

 **N°63-1**

**Simplicité pire**

Test

0135

Taille courte

☐ ☐ ☐ ☒

Lisibilité des variables

☒ ☐ ☐ ☐

Structure compréhensible


☒ ☐ ☐ ☐

**NOTE 5**

## 3.3 - SOBRIÉTÉ

### 3.3.1 - SOBRIÉTÉ MEILLEUR

#### 3.3.1.1 ALGORITHME N°17

 **N°17**

**Sobriété meilleur**

Test

0135

Respect bonne pratique

☐ ☐ ☐ ☒

Lisibilité


☐ ☐ ☐ ☒

Réutilisabilité

☐ ☐ ☐ ☒

**NOTE 15**

#### 3.3.1.2 ALGORITHME N°63

 **N°63**

**Sobriété meilleur**

Test

0135

Respect bonne pratique

☐ ☐ ☐ ☒

Lisibilité

☐ ☐ ☐ ☒


Réutilisabilité

☐ ☐ ☐ ☒

**NOTE 15**

### 3.3.2 - SOBRIÉTÉ PIRE

#### 3.3.2.1 ALGORITHME N°10

 ***n°10***

**Sobriété pire**

Test

0135

Respect bonne pratique

☐ ☐ ☐ ☒

Lisibilité


☐ ☐ ☐ ☒

Réutilisabilité

☒ ☐ ☐ ☐

**NOTE 10**

#### 3.3.2.2 ALGORITHME N°48

 ***n°48***

**Sobriété pire**

Test

0135

Respect bonne pratique

☐ ☐ ☒ ☐

Lisibilité

☐ ☐ ☐ ☒

Réutilisabilité

☒ ☐ ☐ ☐

**NOTE 8**



## 4 - CLASSEMENT DES ALGORITHMES

### CLASSEMENT EFFICACITÉ

TOP 1 : Numéro 48

TOP 2 : Numéro 24

TOP 3 : Numéro 62

~~TOP 4 :~~ Numéro 34 Hors Concours -> Disqualifié

### CLASSEMENT SIMPLICITÉ

TOP 1 : Numéro 40

TOP 2 : Numéro 15

TOP 3 : Numéro 20

TOP 4 : Numéro 44

TOP 5 : Numéro 63-1

### CLASSEMENT SOBRIÉTÉ

TOP 1 : Numéro 17 et Numéro 63

TOP 2 : Numéro 10

TOP 3 : Numéro 48

## CONCLUSION

Cette SAÉ a permis d'approfondir notre réflexion sur l'approche algorithmique des problèmes rencontrés lors des phases de développement. À travers le concours de codage, nous avons eu l'occasion de mettre en pratique nos compétences en résolution de problèmes et en programmation.

Nous avons analysé des critères tels que la lisibilité du code, la qualité du code, l'efficacité, la sobriété numérique et le temps d'exécution. Cette étape nous a confrontés à la diversité des approches et nous a encouragés à évaluer les forces et les faiblesses de chaque solution. La justification objective des comparaisons et du classement a été un aspect essentiel de cette phase.

Grâce à cette SAÉ, nous avons développé nos compétences en programmation, en analyse algorithmique et en évaluation de solutions. Nous avons également pris conscience de l'importance de la clarté du code, de l'efficacité de l'algorithme et de la minimisation de la consommation de ressources.