

## Projet d'algorithmique : Créer un moteur de recherche textuel

### Table des matières

1. Le planning.....	2
2. Répartition des rôles et des tâches .....	3
2.1 Mathias : .....	3
2.2 Juan : .....	3
2.3 Céline : .....	3
3. Outils utilisés.....	4
4. Méthodes et techniques utilisées.....	4
5. Nos spécifications .....	4
6. Notre code .....	5
7. Notre exécutable : test et documentation .....	5
7.1 Voici notre exécutable : .....	5
7.2 Tests : .....	5
7.3 Documentation : .....	7

## 1. Le planning

compréhension du sujet	planification	Création des fichiers cri	recherche du mot choisi dans les différents fichiers	programme principal
compréhension des objectifs demandé	établir un plan des tâches à réaliser	séparer chaque mot les mettre en minuscule enlever les symboles et la ponctuation	réaliser un code permettant de retourner si le mot qu'on cherche et le mot actuel sont les mêmes	créer l'interface ayant accès à un menu
Identifier les différentes étapes logique du moteur de recherche	mettre à jour le plan des tâches au fil de nos avancées	compter le nombre d'occurrences de chaque mot	fonction permettant de trier le nombres d'occurrences par rapport au fichier	pouvoir réaliser la recherche de/ des mot/s que l'on souhaite
éclairer les doutes en posant les questions aux enseignants	finir toutes les tâches établies	créer et remplir les fichiers CRI pour chaque fichier .txt	fonction permettant de faire des recherches sur plusieurs mots	Obtenir un classement des texte les plus pertinents et le nombre d'occurrences du mot en sortie

## 2. Répartition des rôles et des tâches

### 2.1 Mathias :

Mathias a été chargé de créer la fonction `crea_CRI` qui une fois que la fonction de comptage des mots a été créé à dû créer des fichiers CRI et renvoyer cette sortie dans ces fichiers.

Il s'est également chargé du debug des autres fonctions et a tenté d'implémenter des fonctionnalités plus avancées.

Enfin il s'est aussi chargé d'assembler toutes nos fonctions dans un même fichier pour tout tester au même endroit et a séparé le code en plusieurs fichiers.

### 2.2 Juan :

Juan a pour sa part, dû créer la fonction `SearchWord` qui va chercher le mot entré par l'utilisateur dans les CRI et renvoie les textes dans l'ordre le plus pertinent (donc décroissant par nombre d'occurrences du mot).

Création de l'interface ainsi que la présence d'un petit menu pour

Il a également mis en place l'interface d'affichage de notre moteur de recherche et a travaillé avec Céline sur le PowerPoint.

### 2.3 Céline :

Céline a dû créer la fonction qui a permis de compter le nombre d'occurrences dans chaque texte qui a été utilisée dans la fonction `creaCRI`.

Elle a également fait des tests pour l'implémentation de nouvelles idées dans le code et a majoritairement travaillé sur le PowerPoint et le Word.

### 3. Outils utilisés

Afin de réaliser ce projet nous avons utilisés divers outils :

- Nous avons codé en C essentiellement sur l'éditeur de texte de la VMWare de l'école (donc avec un milieu de développement linux) bien qu'il nous soit arrivé de coder sur nos blocs-notes en local.
- Nous avons aussi utilisé Visual Studio Code car on s'est aperçu que cela facilitait l'organisation et la gestion de nos codes.
- Nous avons utilisé Discord comme moyen de communication privilégié car nous habitons loin les uns des autres et il fallait compléter ce projet sur notre temps libre.
- Pour nous partager nos codes, nous avons utilisé GitHub mais aussi discord où on a envoyé des parties du code en fichiers textes.
- Pour notre présentation nous avons utilisé la plateforme Canva et ses modèles qui nous ont semblés le plus pertinents.
- Enfin, pour ce rapport nous avons utilisé Word.

### 4. Méthodes et techniques utilisées

Pour réaliser notre code nous avons essentiellement cherchés des idées sur google mais nous avons aussi demandés des idées de fonctions à nos amis forts en code (en CIN).

### 5. Nos spécifications

Notre moteur de recherche doit réaliser une indexation des différents textes et il doit afficher un classement des textes à la suite d'une recherche d'un ou de plusieurs mots de la part de l'utilisateur.

Les textes doivent être classés par pertinence, c'est à dire que celui où le/les mots à un nombre d'occurrences le plus élevé est le premier. Ils sont donc affichés de manière décroissante.

Il ne doit pas y avoir de différence entre les mots aux singuliers et aux pluriels.

## 6. Notre code

Voici notre code : Vous pouvez retrouver notre code dans l'archive tar.

## 7. Notre exécutable : test et documentation

### 7.1 Voici notre exécutable :

```
Projet_SearchEngine_Jan2023 $ ./main
┌───┬───┬───┐
│ M │ A │ C │
└───┴───┴───┘
What do you want to do ?
0. Quit
1. Do a research
```

Pour Mathias, Atlas, Céline.

### 7.2 Tests :

```
Projet_SearchEngine_Jan2023 $ ./main this
> "this" est présent tant de fois :
Rap_God.CRI : 8
pure_honey.CRI : 5
Network.CRI : 5
Kill_Bill.CRI : 5
Godzilla.CRI : 5
brain.CRI : 5
How_to_be_me.CRI : 2
complain.CRI : 2
Soldier_of_Heaven.CRI : 2
Get_Lucky.CRI : 1
poetry.CRI : 1
gym.CRI : 1
```

```
Projet_SearchEngine_Jan2023 $ ./main this a
```

```
> "this" est présent tant de fois :
```

```
Rap_God.CRI : 8  
pure_honey.CRI : 5  
Network.CRI : 5  
Kill_Bill.CRI : 5  
Godzilla.CRI : 5  
brain.CRI : 5  
How_to_be_me.CRI : 2  
complain.CRI : 2  
Soldier_of_Heaven.CRI : 2  
Get_Lucky.CRI : 1  
poetry.CRI : 1  
gym.CRI : 1
```

```
> "a" est présent tant de fois :
```

```
Rap_God.CRI : 61  
Godzilla.CRI : 40  
Network.CRI : 18  
I'm_not_a_saint.CRI : 12  
gym.CRI : 9  
pure_honey.CRI : 9  
Kill_Bill.CRI : 5  
brain.CRI : 5  
Word_on_Fire.CRI : 4  
poetry.CRI : 2  
cooking.CRI : 2  
guitar.CRI : 1  
How_to_be_me.CRI : 1  
Soldier_of_Heaven.CRI : 1
```

```
Projet_SearchEngine_Jan2023 $ ./main this a

> "this" est présent tant de fois :
Rap_God.CRI : 8
pure_honey.CRI : 5
Network.CRI : 5
Kill_Bill.CRI : 5
Godzilla.CRI : 5
brain.CRI : 5
How_to_be_me.CRI : 2
complain.CRI : 2
Soldier_of_Heaven.CRI : 2
Get_Lucky.CRI : 1
poetry.CRI : 1
gym.CRI : 1

> "a" est présent tant de fois :
Rap_God.CRI : 61
Godzilla.CRI : 40
Network.CRI : 18
I'm_not_a_saint.CRI : 12
gym.CRI : 9
pure_honey.CRI : 9
Kill_Bill.CRI : 5
brain.CRI : 5
Word_on_Fire.CRI : 4
poetry.CRI : 2
cooking.CRI : 2
guitar.CRI : 1
How_to_be_me.CRI : 1
Soldier_of_Heaven.CRI : 1

> "the" est présent tant de fois :
Around_the_world.CRI : 144
Rap_God.CRI : 50
Godzilla.CRI : 49
brain.CRI : 27
pure_honey.CRI : 21
Soldier_of_Heaven.CRI : 17
Network.CRI : 16
Get_Lucky.CRI : 12
gym.CRI : 12
How_to_be_me.CRI : 12
Word_on_Fire.CRI : 10
Kill_Bill.CRI : 8
poetry.CRI : 5
```

## 7.3 Documentation :

Nous avons mis en commentaire, toutes nos explications dans l'archive tar.

