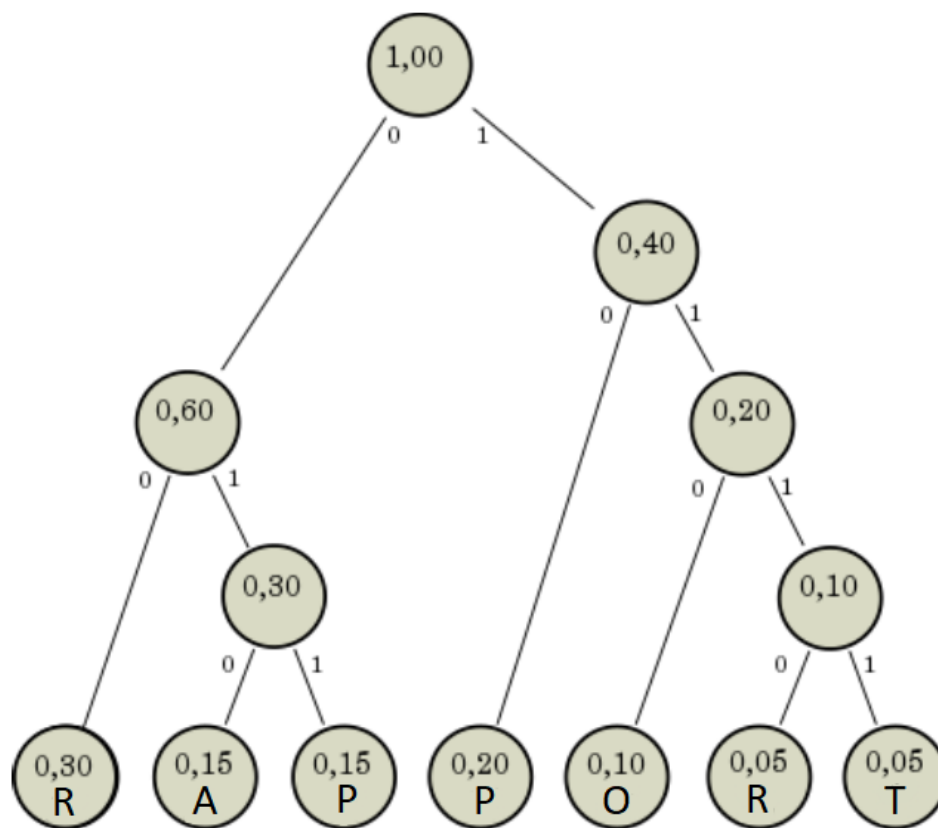


## Algorithme de Huffman (compression d'un fichier texte)



Clément HAUSKNECHT et Walid BOURHALEB

FISA3U31

## Table des matières :

- INTRODUCTION
- CAHIER DES CHARGES
- FAISABILITÉ
- SPÉCIFICATION
  - ◆ Objectif
  - ◆ Méthodes
  - ◆ Diagramme de cas d'utilisation
- CONCEPTION
  - ◆ Diagramme d'activité compression
  - ◆ Diagramme d'activité décompression
- FONCTIONNEMENT
- DIFFICULTÉS
- CONCLUSION

## INTRODUCTION :

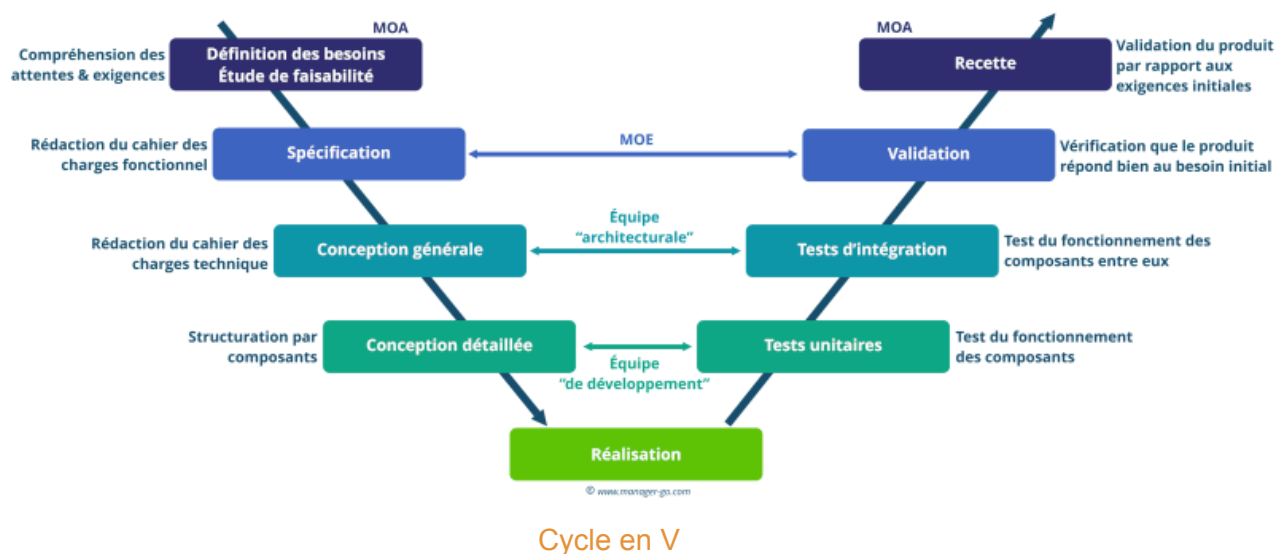
Dans ce rapport, nous allons nous intéresser à la compression d'un fichier à l'aide de l'algorithme de Huffman. Pour se faire, nous avons constitué une petite équipe avec comme personnes concernées Clément HAUSKNECHT et Walid BOURHALEB. L'objectif de ce projet est d'obtenir un programme pouvant compresser des fichiers textes sans pertes d'informations afin qu'ils prennent moins de place sur le disque de l'ordinateur en question.

## CAHIER DES CHARGES :

- Créer un code en langage C permettant la compression d'un fichier texte
- Faire ce projet dans le temps qui nous a été donné qui est de 51 jours
- Concevoir le code à partir de l'algorithme de Huffman ; c'est-à-dire de respecter les propositions de prototypage

## FAISABILITÉ :

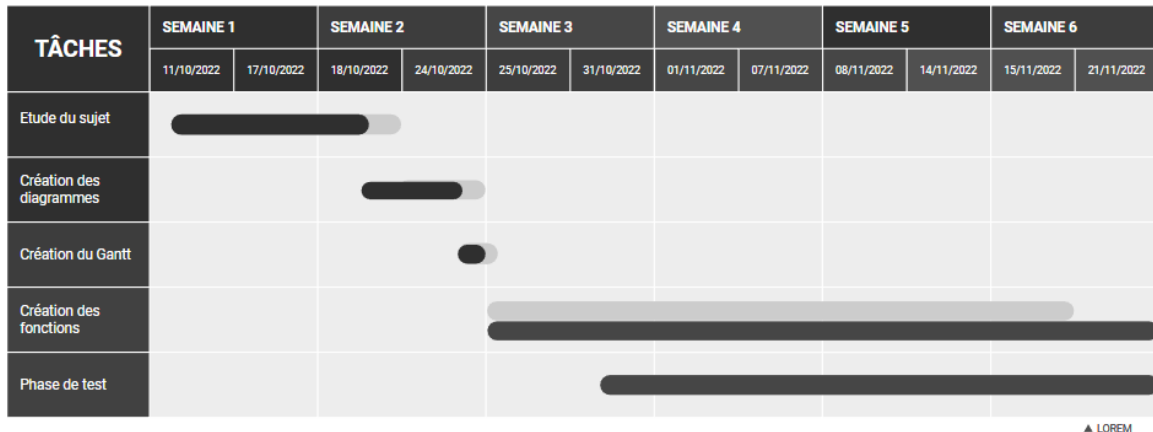
Pour mener à bien ce projet en langage C, nous avons structuré le projet en différentes étapes divisé en deux phases : la première qui est la phase de vérification, puis la seconde qui est la phase de validation. Voici ci-dessous une image du cycle en V :



Avant de commencer la partie spécification du projet, on a dans un premier temps étudié la faisabilité de celui-ci. L'analyse du sujet nous a permis de constater que nous avons les compétences nécessaires pour mener à bien ce projet dans le temps imparti qui est de 51 jours.

Pour que l'on s'organise et que l'on perde le moins de temps possible, nous nous sommes réparties les tâches sur toute la période en nous basant sur notre estimation et des recherches sur la compression de Huffman.

Voici ci-dessous le diagramme de Gantt :



## SPÉCIFICATION :

### Objectif :

L'algorithme de Huffman consiste à recoder les octets d'un ensemble de données source avec des valeurs de longueur binaire variable. Le traitement va se faire en bit car en effet, l'algorithme de Huffman a pour principe de recoder les données qui ont une occurrence très faible sur une longueur binaire élevée. Il va aussi recoder les données très fréquentes sur une longueur binaire très courte.

### Méthodes :

**1 - Algorithme de huffman :** Recoder les valeurs des données, ici les lettres de l'alphabet afin que les lettres les plus fréquemment utilisées aient une longueur binaire plus courte. Ainsi, il y a une compression de bits pour chaque répétition.

**2 - Code-préfixe :** Permet d'éviter l'ambiguïté au moment du décodage

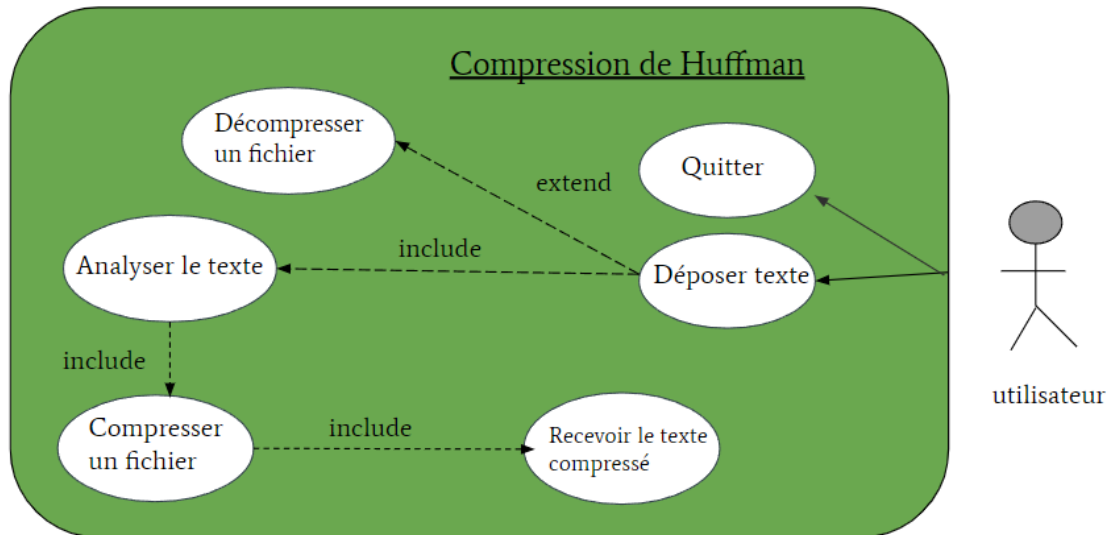
**3 - Arbre binaire :** Sert à affecter pour chaque lettre une valeur en bits

Ensuite, on crée un tableau de fréquence pour chaque lettre comme ceci :

Clé	Valeur
0	Fréquence(0)
1	Fréquence(1)
...	
255	Fréquence(255)

## Diagramme de cas d'utilisation :

Dans cette partie, on va établir notre diagramme de cas d'utilisation. A savoir que ce diagramme modélise le comportement d'un système et permet de capturer les exigences de celui-ci. Il s'occupe de décrire les fonctions générales et la portée du système. Voici ci-dessous la représentation de notre diagramme :



Dans le diagramme de cas d'utilisation de la compression de Huffman nous constatons qu'il y a 1 acteur qui est l'utilisateur.

L'utilisateur dispose de plusieurs choix (actions) qui lui sont proposés :

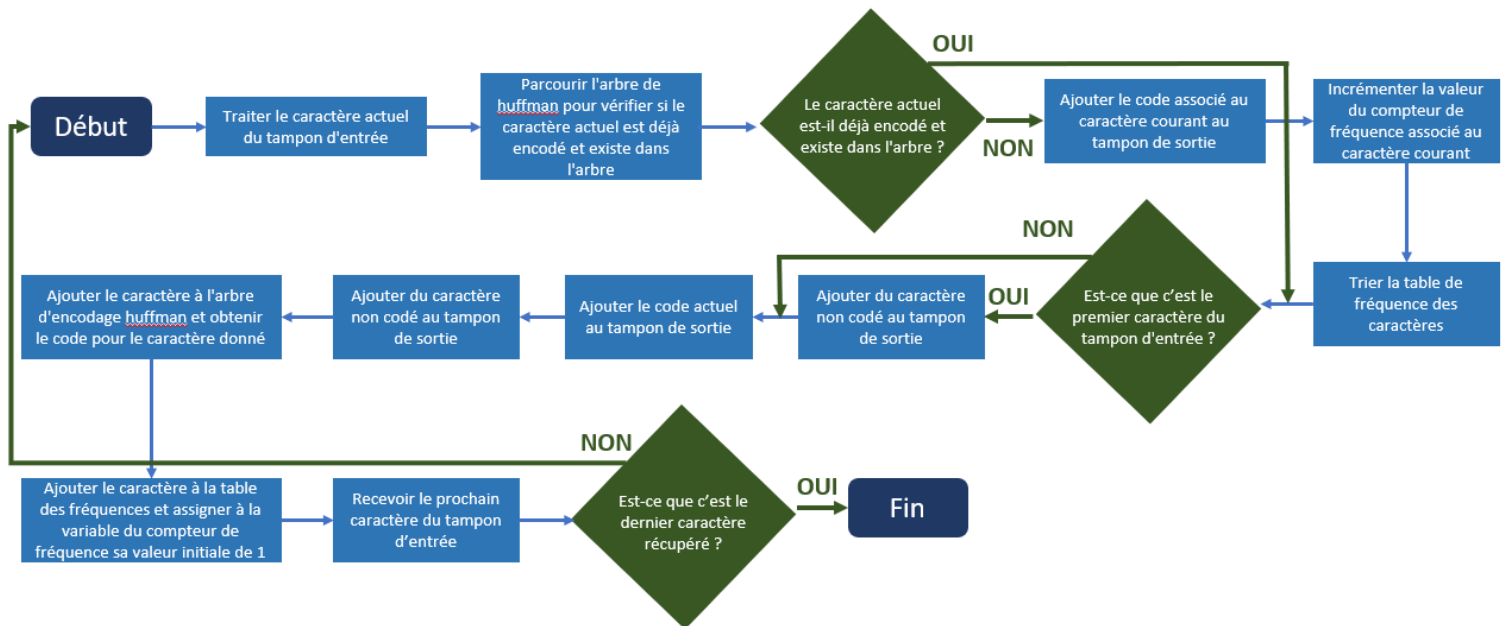
- Si il choisit de compresser ou décompresser un texte, il doit d'abord déposer le texte, puis le fichier est analysé pour être compressé ou décompressé, et enfin il récupère le texte sous le format correspondant.
- Si il décide de quitter, le programme s'arrête tout simplement.

Vous pouvez ajouter des titres (Format > Styles de paragraphe) qui apparaîtront dans votre table des matières.

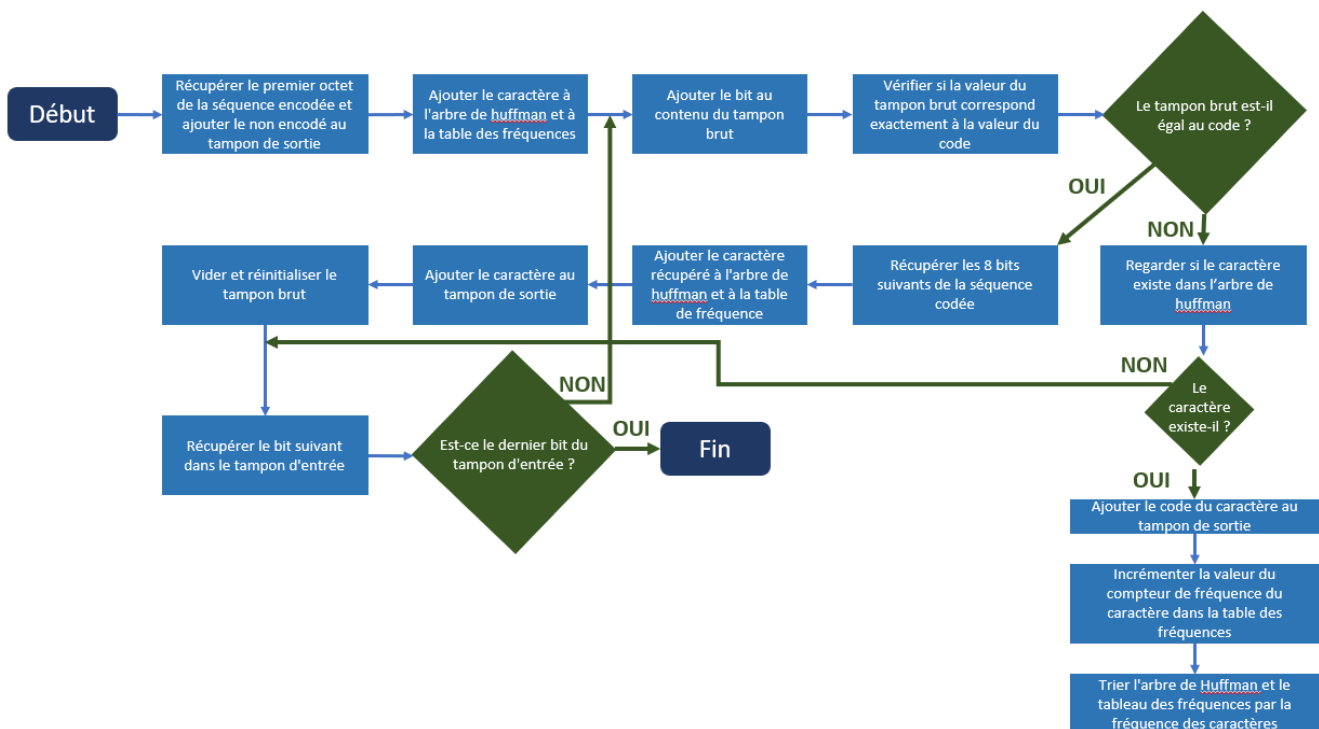
## CONCEPTION :

Dans cette partie destinée à la conception, il est nécessaire de faire dans un premier temps le diagramme d'activité qui va nous servir d'appui lors de la création de notre code. Le diagramme d'activité a pour utilité de nous fournir la vue du comportement de notre code en décrivant les différentes séquences d'actions. Il va par la même occasion nous montrer les flux entre les différentes actions du code.

### Diagramme d'activité compression :



### Diagramme d'activité de décompression :



## FONCTIONNEMENT :

Notre programme se divise en plusieurs fonctions principales :

### TRI DU TABLEAU

Tri le tableau de caractères en fonction de leurs occurrences

### ARBRE HUFFMAN

Construction de l'arbre de Huffman d'après un tableau de feuille avec les nœuds en commençant par les caractères avec le plus d'occurrences.

### PARCOURIR

Conversion du nouveau code en binaire, écriture dans un fichier temporaire.

### ENCODAGE

Pour chaque caractère du fichier à compresser, écriture du nouveau code dans un fichier temporaire

### FINALISATION

Écriture du texte final compressé et du dictionnaire dans le fichier compressé

## DIFFICULTÉS :

Concernant les difficultés que nous avons rencontrées, nous avons notés notamment :

- Malgré que notre programme fonctionne pour la compression, ce n'est pas le cas pour la décompression. Nous n'avons pas eu le temps de le rendre fonctionnel.
- Nous avons trouvé que l'algorithme de Huffman était compréhensible, mais que le code était difficile à effectuer. En effet, nous avons surestimé le temps consacré au développement du programme.
- Enfin, l'organisation difficile entre l'entreprise, l'école et les activités périscolaires.

## CONCLUSION :

Pour conclure, nous avons réussi à compresser les bits d'un fichier .txt afin de diminuer le poids de ce fichier .txt.

Malgré que la partie décompression ne soit pas encore fonctionnelle, nous sommes satisfait de notre résultat. En effet, au commencement du projet nous avons effectué énormément de recherches car il nous manquait beaucoup de connaissances liées à la programmation en C. C'était donc un challenge qui nous a permis d'en apprendre à la fois sur le langage C mais aussi la programmation en général.

De plus, ce projet nous a permis de mettre à l'épreuve nos connaissances théoriques en langage C sur un exercice concret.