

## RAPPORT

### Sommaire :

I - Fonctionnement du programme.

II - Bugs éventuels.

### I - Fonctionnement du programme

#### Chiffrement :

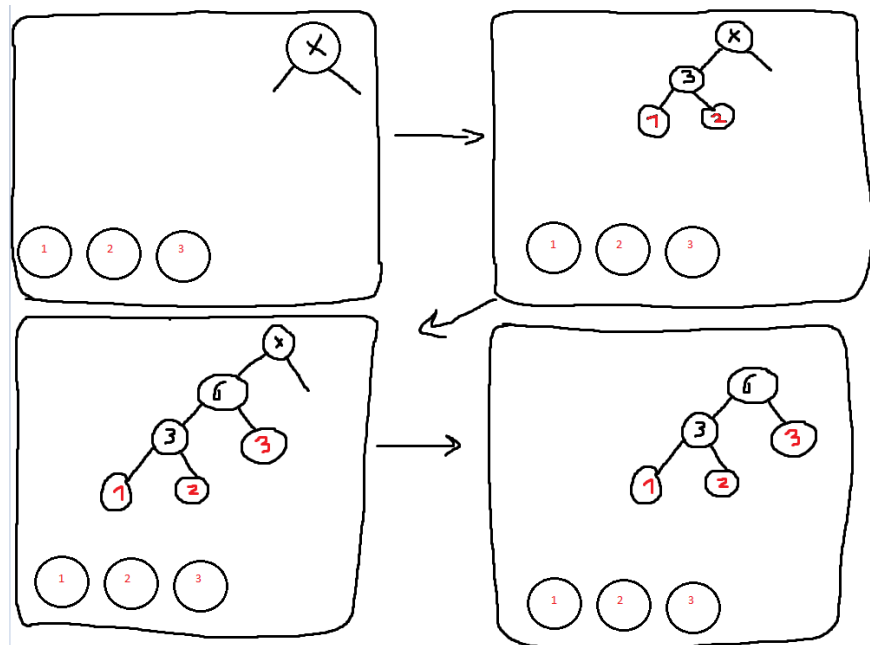
-Récupération des données.

Après avoir récupéré le fichier texte, le programme va créer un tableau de structure. Chaque élément de la structure contient un entier correspondant au chiffre ascii d'un caractère ainsi qu'un entier correspondant au nombre d'occurrences du caractère en question. une fois le tableau rempli le programme va alors le trier par ordre croissant d'occurrence.

-Création de l'arbre de codage huffman.

Commence ensuite la création de l'arbre. Pour créer cet arbre il faut commencer par le bas de l'arbre, les feuilles. Nous avons donc décidé, pour pouvoir garder un pointeur sur le plus haut point de l'arbre, de créer un nœud "père" ou l'on va implémenter notre vrai arbre dans un des fils. Ce n'est pas l'option la plus optimisée mais c'est la première solution qui nous est apparue sur le moment.

Exemple :



## -Chiffrement.

Nous avons décidé de stocker premièrement notre texte chiffré dans un .txt que nous déplacerons après dans un .bin.

Le fichier .bin final sera composé de la sorte :

- Premier octet : le nombre de 0 à supprimer à la fin.\*
- Deuxième octet : la taille du tableau.
- Troisième octet : précisant comment lire le tableau dans le fichier .bin.\*\*
- Le tableau de caractère et d'occurrence.
- Le texte chiffré.

*\*Les caractères du fichier .txt créé précédemment ( composé de 0 et de 1 étant les caractères chiffré par le codage huffman) vont être regroupés par paquets de 8 ( un octet ) puis converti en un entier pour être ensuite envoyé dans le .bin. Si il n'y a pas assez de bit pour remplir le dernier octet alors il va être rempli avec autant de 0 que nécessaire.*

*\*\*Si aucun caractère n'apparaît plus de 256 fois alors les occurrences du tableaux seront inscrites sur 1 octet, au contraire si un caractère apparaît plus de 256 fois il faut inscrire les occurrences sur 2 octet.*

## Déchiffrage:

Le déchiffrage se déroule de la même manière que le chiffage mais en sens inverse. Une fois le tableau récupéré, l'arbre est recréé de la même façon. Le texte chiffré est remis dans un fichier temporaire .txt ( par expérience si le texte est trop long, mettre le texte chiffré dans une liste chaînée ou un tableau pourrait poser des problèmes de place ). Ensuite le fichier texte est lu caractère par caractère ( 0 ou 1 ) pour déchiffrer le texte grâce à l'arbre.

## III - Bugs éventuels.

-Il peut arriver que des caractères au début soient supprimés dans le fichier texte déchiffré.

-L'affichage de certains caractères spéciaux peuvent parfois être défectueux dans le texte déchiffrer. Après comparaison octet par octet du fichier original et du fichier texte déchiffré, les caractères spéciaux sont correctement récupérés. Ainsi, le problème est causé par le lecteur de fichier texte.