

# Rapport : projet LDP parti 1.

Talhaoui Ahmed, Matricule : 568936

March 17, 2023

## 1 Lancement du code :

Pour lancer le programme, il vous suffit de lancer la commande « make » sur un terminal linux, le lancement avec les fichiers « utils.cpp » et « utils.hpp » est impératif au bon fonctionnement du programme, car ces fichiers ont été modifié et certaines fonctions ont été ajoutées dans ces fichiers afin de ne pas nuire à la lisibilité du programme et ainsi faciliter les futures modifications potentielles du code.

Ainsi vous générerez, deux fichiers "encodeur" et "decodeur" que vous pourrez exécuter avec la commande suivante :

```
./« nomDuFichier » .
```

veuillez également mettre les fichiers .txt dans le bon ordre à l'exécution du programme :

- pour l'encodeur : message.txt arbre.txt messageCoder.txt
- pour le decodeur: messageCoder.txt arbre.txt message.txt

## 2 Différence code python et c++ :

Les principales différences entre le code python et c++, est que en c++, beaucoup de fonction supplémentaire permettant certaines actions ont été ajoutées, par exemple dans le fichier « utils.cpp » multiples fonctions tel que « getlenchar » par exemple ont été écrites afin de permettre certaines actions qui ne sont pas disponibles par défaut en c++ mais pourtant triviale en python. L'utilisation des pointeurs est également une différence majeure par rapport au code python qui ne fonctionne que par référence.

Mis à part cela, les codes des deux langages restent très similaires.

## 3 Variable statique :

L'avantage des variables statiques est qu'elles permettent d'économiser de l'espace et des nouvelles initialisations de variables, car elles n'ont besoin que d'être déclarées qu'une seule fois pour pouvoir être utilisées à l'échelle globale dans le code. En revanche elles nuisent à la robustesse du programme car un changement involontaire de la valeur de la variable par exemple sera source de nombreux bugs potentiels et difficile à déboguer.

Une alternative serait de stocker ses données sur un « heap » au lieu d'une « Zone statique ». Cela reviendrait à allouer dynamiquement de l'espace à une variable. Cela a comme inconvénient que le programmeur ait à devoir gérer manuellement la réservation d'espace mémoire, mais ça permet donc aussi d'optimiser l'espace utilisé d'une variable. De plus la fiabilité du programme sera de meilleure qualité puisque ce genre d'allocation permet d'avoir un meilleur contrôle du scope d'une variable, ce qui agrége donc de pouvoir réutiliser un même nom de variable pour des contextes complètement différents dans une même partie de code. Pour les différentes raisons abordées, l'allocation dynamique semble donc préférable aux variables statiques.