

Suite du cryptage

Le but est maintenant de créer un programme qui crypte un fichier.

Partie 1 : crypter un fichier texte

Faites la fonction suivante :

```
void cryptFichier(string nomFichier)
```

Le premier paramètre est le nom du fichier texte à ouvrir pour crypter. La fonction lira le fichier ligne par ligne et enregistrera chacune des lignes cryptées dans un autre fichier (que vous appellerez « crypt.txt »).

Partie 2 : décrypter un fichier texte

Faites la fonction suivante

```
void decryptFichier(string nomFichier)
```

Le premier paramètre est le nom du fichier texte à ouvrir pour décrypter. La fonction lira le fichier ligne par ligne et enregistrera chacune des lignes décryptées dans un autre fichier (que vous appellerez « decrypt.txt »).

Partie 3 : crypter un fichier quelconque.

Pour travailler avec des fichiers binaires (exe, images, zip...) le principe est le même mais la méthode pour ouvrir les fichiers est un peu différente. Commençons donc par ouvrir :

Pour ouvrir en lecture un fichier binaire, il suffit d'appeler le constructeur qui a en paramètre le nom du fichier et le mode d'ouverture et fixer ce deuxième paramètre à `ios::in | ios::binary`: par exemple `ifstream f("toto.xyz",ios::in | ios::binary);`

Pour ouvrir en écriture un fichier binaire, il suffit d'appeler le constructeur qui a en paramètre le nom du fichier et le mode d'ouverture et fixer ce deuxième paramètre à `ios::out | ios::binary`: par exemple `ofstream fichier("toto.zip",ios::out | ios::binary);`

Pour lire dans un fichier binaire, on ne peut plus utiliser `getline` puisqu'il n'y a pas de ligne...

On utilisera la fonction `read((char *)buffer , int nb)` pour lire nb octets de ce fichier. **Buffer** peut être un tableau de `char` et **nb** correspondra au nombre de cases du tableau.

On utilisera la fonction `write` un peu de la même manière. Je vous laisse le soin de chercher sur internet comment utiliser ces fonctions.

Modifiez les fonctions crypter et décrypter pour pouvoir travailler avec des tableaux de char plutôt que des string.