

# CAPES MATHS OPTION INFORMATIQUE

## COMPRESSION ET CORRECTION

### Compression RLE

Soit un fichier pouvant contenir des séquences répétitives d'entiers, par exemple {12, 6, 6, 6, 6, 6, 1, 3, 8, 8, 10, 2, 53, 53, 53, 53, 6, 6, 6, 13}.

Une technique de compression consiste à écrire pour chaque entier le nombre de répétitions supplémentaires. On pourrait donc remplacer {53, 53, 53, 53, 53} par {53, 4}, ce qui signifie : l'entier 53, suivi d'encore 4 fois l'entier 53. Le problème de cette idée est que la séquence {53, 53, 53, 53, 53} va bien être compressée, mais qu'au contraire, toutes les séquences non répétitives vont être dilatées : par exemple, {10} serait réécrit en {10, 0}.

Une meilleure idée consiste à ne transformer une séquence que si un entier est immédiatement répété au moins une fois. Ainsi, {53, 53, 53, 53, 53} serait réécrite en {53, 53, 3}, {10} serait inchangée et {8, 8} serait réécrite en {8, 8, 0}. La séquence donnée en exemple serait donc réécrite de la façon suivante : {12, 6, 6, 3, 1, 3, 8, 8, 0, 10, 2, 53, 53, 3, 6, 6, 1, 13}. Lors de la décompression, c'est le fait d'avoir deux entiers identiques à la suite qui indique que l'entier suivant est un nombre d'occurrences supplémentaires et non un entier du fichier de départ. **C'est à cette seconde idée que vous allez vous intéresser dans cet exercice.** Il s'agit de la compression dite « RLE » (Run Length Encoding). Notez que bien que ce second algorithme soit la plupart du temps meilleur que le premier, il n'est pas parfait pour autant : il existe tout de même des cas pour lesquels il dilate au lieu de compresser. Il favorise les successions de plus de trois caractères identiques, il défavorise les successions de deux et laisse inchangé les simples occurrences.

- Supposons que le fichier d'entrée contient 10 entiers. Donnez un exemple de contenu de fichier qui donnerait la plus petite taille possible pour le fichier de sortie, puis un exemple de contenu qui donnerait la plus grande taille possible en sortie.
- Donner le code Python de la procédure de compression, dont l'entête est la suivante :

```
def compresser (nomFichierEntree, nomFichierSortie)
# Préconditions : nomFichierEntree est le nom d'un fichier contenant une séquence
# d'entiers
# Postconditions : un nouveau fichier nommé comme spécifié dans nomFichierSortie est
# créé, son contenu correspond à la séquence d'entiers compressée par l'algorithme RLE
```

- Donner le code Python de la procédure de décompression, dont l'entête est la suivante :

```
def decompresser (nomFichierEntree, nomFichierSortie)
# Préconditions : nomFichierEntree est le nom d'un fichier compressé selon l'algo RLE
# Postconditions : un nouveau fichier nommé comme spécifié dans nomFichierSortie est
# créé, son contenu correspond à la décompression du fichier d'entrée
```