

Nous souhaitons écrire un programme java (la classe `CompresseurTexteEtudiant`) permettant de compresser des fichiers textes selon la méthode de codage par plages (**Run Length Encoding**). Le programme utilisera deux formats d'encodage : le format A (extension `.txtA`) et le format B (extension `.txtB`). Il prendra deux arguments :

- le premier argument est une chaîne de caractères permettant d'identifier le dossier dans lequel se trouveront les fichiers à traiter,
- le deuxième argument sera une des 4 chaînes de caractères suivantes : `convertA`, `convertB`, `afficheA`, `afficheB`. Dans le cas où le deuxième argument est `convertA` (resp. `convertB`), le programme générera dans le dossier spécifié des fichiers d'extension `.txtA` (resp. `.txtB`) correspondant aux fichiers `.txt` trouvés dans le dossier et convertis à l'aide du format A (resp. B). Lorsque le deuxième argument est `afficheA` (resp. `afficheB`), le programme doit décoder les fichiers portant l'extension `.txtA` (resp. `.txtB`) et afficher leur contenu à l'écran.

### **La compression de type Format A**

La compression se fait ligne par ligne. Chaque séquence maximale d'un même caractère `car1` (même pour les séquences de longueur une) ayant une longueur maximale de 26 est encodé par les deux caractères `car2car1` où `car2` représente le nombre d'occurrences de `car1`. `car2` est plus précisément le caractère entre 'a' et 'z' tel que `car2` est 'a'+(nombre d'occurrences de `car1` – 1).

Ainsi la ligne : `AAAAAAAAABBBBBBBCDDDE` sera encodée par :  
`hAhBaCzDmDaEaF`

**Question 1.** Récupérez le fichier `dossierEtudiantSuperTexte.zip` se trouvant sur la plateforme Moodle et décompressez-le. Incorporez le fichier `CompresseurTexteEtudiant.java` dans un nouveau projet eclipse (générez la doc correspondant à cette classe). Complétez les zones (0), (1), (2), (3), (4) du fichier `CompresseurTexteEtudiant.java` puis, lancez le programme afin de convertir les fichiers `.txt` au format A se trouvant dans le dossier `exemplesTextesSimples`. Vérifiez que l'encodage est correct. En comparant les tailles des fichiers générés avec les fichiers sources que constatez-vous ?

**Question 2.** Complétez les zones (5), (6), (7) du fichier `CompresseurTexteEtudiant.java` puis, lancez le programme afin d'afficher le décodage des fichiers encodés précédemment au format A.

### **La compression de type Format B**

À l'aide de vos réponses à la question 1, vous vous êtes aperçu que la compression de type Format A n'était pas intéressante dans le cas où le texte à encoder ne contenait pas de nombreuses séquences d'un même caractère avec une longueur suffisamment élevée. Afin d'améliorer la compression nous introduisons la compression de type Format B. Avec ce nouvel encodage, seules les séquences d'un même caractère avec une longueur supérieure ou égale à 3 seront encodées de la même manière que pour l'encodage de type A. Pour les autres séquences nous utiliserons le principe suivant. Une séquence `seq` d'au plus 26 caractères est extraite, puis cette séquence est encodée avec la chaîne de caractères composé du caractère `car` suivi de la séquence `seq` où `car` est le caractère correspondant à 'A'+(nombre de caractères de `seq` – 1).

Ainsi la ligne :

Bonjour comment allez-vous Monsieur `AA`  
`BBBBBCCCCCD` !?

sera encodée par :

`ZBonjour comment allez-vousK Monsieur zAmAA eBeCED` !?

**Question 3.** Complétez les zones (8), (9), (10), (11) du fichier `CompresseurTexteEtudiant.java` puis, lancez le programme afin de convertir les fichiers `.txt` se trouvant dans le dossier `exemplesTextesSimples` au format B. Vérifiez que l'encodage est correct. En comparant les tailles des fichiers générés avec les fichiers sources et les différents formats que constatez-vous ?

**Question 4.** Complétez les zones (12), (14), (15) du fichier `CompresseurTexteEtudiant.java` puis, lancez le programme afin d'afficher le décodage des fichiers encodés précédemment au format B.

**Question 5.** Convertissez les fichiers textes se trouvant dans le dossier `exemplesTextesPlus` en utilisant le format B. Lancez l'application java `superTexteAfficheur1.0.jar` qui permet d'afficher du texte au format B. C'est cool ! non ?

