R1.01 : Initiation au développement (partie 2)
Feuille TD n° 7

Fichiers texte (= Fichiers éditables) avec contenus simples

Objectifs :
1.- Apprendre les manipulations de base d'un fichier de texte.
2.- Appliquer les algorithmes classiques sur un fichier de texte

1.- Algorithmes sur un fichier (de lignes) de texte

Les fichiers considérés dans cet exercice sont des fichiers de texte constitués de chaînes de caractères séparées entre elles par un saut de ligne (caractère '\n').

On considère les fichiers donnés corrects, c'est à dire éventuellement vides, mais sans erreur. On souhaite écrire 3 sous-programmes indépendants :

A.- Sous-programme **`afficherFichierTexte`** qui affiche à l'écran le contenu d'un tel fichier de texte dont le nom système est fourni en paramètre.

B.- Sous-programme **`etendreFichierTexte`** qui ajoute le contenu d'un fichier de texte (le fichier source) à la fin (= après le dernier enregistrement) d'un autre fichier de texte (le fichier cible). Les noms système des fichiers concernés sont fournis en paramètre.

C.- Sous-programme **`afficherInverseFichierTexte`** qui affiche à l'écran, *dans l'ordre inverse*, le contenu d'un fichier texte dont le nom système est fourni en paramètre.

Exemple :

Si contenu du fichier =

```
Ceci est la ligne 1
Ceci est la ligne 2
Ceci est la ligne 3
Ceci est la ligne 4
```

à l'écran :

```
Ceci est la ligne 4
Ceci est la ligne 3
Ceci est la ligne 2
Ceci est la ligne 1
```

Travail à faire

Pour chacun des sous-programmes :

1) Écrire la déclaration du sous-programme (entête + BUT !)
2) Identifier le modèle d'algorithme vu en cours permettant de réaliser le but donné. Préciser la stratégie.
3) Écrire l'algorithme réalisant le but donné.

2.- Fichier de nombres éditable : Calcul de moyennes pluviométriques

On considère un ensemble de valeurs numériques entières stockées dans un fichier texte, séparées par un séparateur habituel (saut(s) de ligne et/ou tabulation(s) et/ou espace(s)). Ces valeurs correspondent à des relevés pluviométriques réalisés tous les jours par une station météorologique durant un mois, à raison de 1 relevé par jour. Les relevés sont consignés en millimètres.

On considère le fichier correct, c'est à dire non vide et contenant toujours autant de valeurs qu'il y a de jours dans le mois correspondant, **et** sans erreur.

Chaque fichier a pour nom système le mois et l'année où ont eu lieu les relevés (exples : dec-2014, fev-2015).

On souhaite écrire un sous-programme **`moyennePluviometrique`** qui calcule et retourne la quantité moyenne d'eau tombée durant 1 mois à partir des valeurs consignées dans un fichier dont le nom système est fourni en paramètre.

Travail à faire

1) Écrire la déclaration du sous-programme
2) Identifier le modèle d'algorithme vu en cours permettant de réaliser le but donné. Préciser la stratégie.
3) Écrire l'algorithme réalisant le but donné.

3.- Fichier de caractères : Cryptage

A.- version simple

On souhaite écrire un sous-programme **`cryptage`** qui, à partir d'un fichier texte dont le nom système est fourni en paramètre, créée un autre fichier texte où toutes les voyelles ont été transformées selon l'algorithme de cryptage suivant : 'a' → 'e', 'e'→ 'i', 'i' → 'o', 'o' → 'u' et 'u' → 'a', qu'il s'agisse de minuscules ou de majuscules.

Le nom système du fichier crypté sera généré par le sous-programme à partir du nom système fourni, par exemple en ajoutant la chaîne "_crypted" au nom système fourni (avant l'extension si elle existe).

Exemples :		
	lettreAnonyme.txt	→ lettreAnonyme_crypted.txt
	lettreAnonyme	→ lettreAnonyme_crypted
	module.h	→ module_crypted.h
	.h	→ .h_crypted
	../x	→ ../x_crypted
	../x.cpp	→ ../x_crypted.cpp

Travail à faire

1) Écrire la déclaration du sous-programme **`cryptage`**
2) Identifier le modèle d'algorithme vu en cours permettant de réaliser le but donné. Préciser la stratégie.
3) Écrire l'algorithme réalisant le but donné
4) Écrire la déclaration du sous-programme chargé de générer le nouveau nom système à partir de celui qui lui est fourni

B.- version étendue

Généraliser le problème en considérant que :

- Tout caractère peut être crypté
- On fournit au sous-programme le nom d'un second fichier contenant les correspondances entre caractères cryptés en caractères en clair.
  La forme de ce fichier sera la suivante, en prenant comme exemple de cryptage celui présenté au point A.- :

```
a  e
e  i
i o
o  u
u a
A  E
E I
I  O
O U
U  A
```

- Un binôme 'caractère en clair → 'caractère crypté par ligne
- Le caractère en clair et le caractère crypté sont séparés par au moins une espace
- Ce fichier des correspondances est supposé être **sans erreur** : chaque ligne contient exactement 2 caractères séparés par au moins une espace.

- Les caractères ne figurant pas dans le fichier de correspondances seront retranscrits à l'identique.
- Le reste du comportement du programme ne change pas par rapport à la version simple A.-.

Travail à faire

1) Écrire la déclaration du sous-programme **`cryptage`**
2) Identifier le modèle d'algorithme vu en cours permettant de réaliser le but donné.
3) Écrire l'algorithme réalisant le but donné