

TD8 : éléments de correction

Exercice 2

Fonction : **DonnerIndice**

Précondition : c appartient à l'ensemble $\{'a', \dots, 'z', ' '\}$

Entrée : - c : caractère

Local : /

Sortie : - entier

Début

Si $c == ' '$ **Alors**

 Retourner 26

Fin Si

 Retourner $ASCII(c) - ASCII('a')$ // donne le code ASCII du caractère, ex : $ASCII('a') = 97$

Fin

Fonction : **Code2Gramme**

Précondition : message ne contient que des caractères appartenant à l'ensemble $\{'a', \dots, 'z', ' '\}$

Entrée : - grille : tableau 2D de 27×27 chaînes de caractères

 - message : chaîne de caractères

Local : - indice1, indice2 : entier

Sortie : - messageCode : chaîne de caractères

Début

 Soit messageCode = ""

 Soit indice1 = -1

 Soit indice2 = -1

Pour i **de** 0 **à** Longueur(message) - 1 **avec un pas de** +2 **Faire** // lecture des caractères 2 par 2

 Soit indice1 = **DonnerIndice**(message[i])

 Soit indice2 = **DonnerIndice**(message[i + 1])

 Soit messageCode = messageCode + grille[indice1, indice2]

Fin Pour

 Retourner messageCode

Fin

Fonction : **DonnerCaractere**

Précondition : indice appartient à l'intervalle [0, 26]

Entrée : - indice : entier

Local : /

Sortie : - caractère

Début

Si indice == 26 Alors

Retourner ' '

Fin Si

Retourner **Caractere**(indice + ASCII('a')) // conversion en caractère, ex : *Caractere*(97) = 'a'

Fin

Procédure : **RechercheDansGrille**

Précondition : le message ne contient que des caractères appartenant à l'ensemble {'a', ... 'z', ' '}

Entrée : - grille : tableau 2D de 27*27 chaînes de caractères

- message : chaîne de caractères

- indice1 : entier (modifiable)

- indice2 : entier (modifiable)

Local : /

Sortie : void

Début

Soit indice1 = -1

Soit indice2 = -1

Pour i de 0 à 26 avec un pas de +1 Faire

Pour j de 0 à 26 avec un pas de +1 Faire

Si grille[i, j] est égale à message Alors

Soit indice1 = i

Soit indice2 = j

Retourner // sortir de la procédure

Fin Si

Fin Pour

Fin Pour

Fin

Fonction : **Decode2Gramme**

Précondition : c appartient à l'ensemble {'a', ... 'z', ' '}

Entrée : - grille : tableau 2D de 27*27 chaînes de caractères

- message : chaîne de caractères

Local : - indice1, indice2 : entier

Sortie : - messageDecode : chaîne de caractères

Début

Soit messageDecode = ""

Soit indice1 = -1

Soit indice2 = -1

Pour i de 0 à Longueur(message) - 1 avec un pas de +3 Faire // lecture des caractères 3 par 3

Soit msg = **SousChaine**(message, i, 3) // sous-chaîne de message de l'indice i à i+3 (*)

RechercherDansGrille(grille, msg, indice1, indice2) // modifie indice1 et indice2 !

Si indice1 ≠ -1 ET indice2 ≠ -1 Alors

Soit messageDecode = messageDecode + **DonnerCaractere**(indice1)

+ **DonnerCaractere**(indice2)

Fin Si

Fin Pour

Retourner messageDecode

Fin

(*) en C# on utilisera la fonction Substring() qui s'applique sur une chaîne, prend en paramètre la position de départ et le nombre de caractère à extraire depuis cet indice, et retourne la sous-chaîne correspondante.

Ex :

string message = "hello world !"

string s = message.Substring(6, 5) // s vaut "world"

Procédure : **Main**

Entrée : **void**

Local :
- grille : **tableau 2D** de 27*27 **chaînes de caractères**
- messageU, messageUCode, messageUDecode : **chaîne de caractères**

Sortie : **void**

Début

Soit grille = **tableau 2D** de 27*27 **chaînes de caractères** initialisée selon l'énoncé

Soit messageU = **SaisirChaîne**()

Soit messageUCode = **Code2Gramme**(grille, messageU)

Afficher(messageUCode)

Soit messageUDecode = **Decode2Gramme**(grille, messageUCode)

Afficher(messageUDecode)

Fin