

## TD8 : éléments de correction

### Exercice 3

Fonction : **CreerGrille**

Précondition :  $\text{alphabet} > 0$

Entrée :  
-  $\text{alphabet}$  : entier  
-  $\text{indiceLigne1}$  : entier  
-  $\text{indiceColonne1}$  : entier

Local : /

Sortie : - grille : tableau 2D de  $\text{alphabet} * \text{alphabet}$  chaînes de caractères (de taille 3 chacune)

Début

```
Soit grille = tableau 2D de  $\text{alphabet} * \text{alphabet}$  chaînes de caractères initialisées à ""
Soit i = indiceLigne1
Soit j = indiceColonne1
Soit grille[i, j] = ChaineSur3Caractères(1) // convertit l'entier 1 en la chaîne "001"
Pour k de 2 à  $\text{alphabet} * \text{alphabet}$  Faire
    Soit inext = i - 1
    Soit jnext = j + 1
    // dépassement vers la droite
    Si jnext ==  $\text{alphabet}$  Alors
        Soit jnext = 0
    Fin Si
    // dépassement vers la gauche
    Si inext < 0 Alors
        Soit inext =  $\text{alphabet} - 1$ 
    Fin Si
    // case déjà occupée
    Si grille[inext, jnext] n'est pas égale à "" Alors
        Soit inext = i + 1
        Soit jnext = j
        Si inext ≥  $\text{alphabet}$  Alors
            Soit inext = 0
        Fin Si
    Fin Si
    // cas général
    Soit grille[inext, jnext] = ChaineSur3Caractères(k)
    Soit k = k + 1
    Soit i = inext
    Soit j = jnext
Fin Pour
Retourner grille
```

Fin

Fonction : **DonnerIndice**

Précondition :  $c$  appartient à l'ensemble  $\{'A', \dots, 'Z', ' ', '.'\}$

Entrée : -  $c$  : caractère

Local : /

Sortie : - entier

Début

```
    Si  $c == ' '$  Alors
    |   Retourner 26
    Sinon Si  $c == '.'$  Alors
    |   Retourner 27
    Sinon
    |   Retourner  $ASCII(c) - ASCII('A')$     // conversion code ASCII  $\rightarrow$  caractère
    Fin Si
```

Fin

Fonction : **Code2Gramme**

Précondition :  $alphabet > 0$

Entrée : - grille : tableau 2D de  $alphabet * alphabet$  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, 27]

Entrée : - indice : entier

Local : /

Sortie : - caractère

Début

Si indice == 26 Alors

Retourner ''

Sinon Si indice == 27 Alors

Retourner ''

Sinon

Retourner Caractere(indice + ASCII('A')) // conversion code ASCII → caractère

Fin Si

Fin

Procédure : **RechercheDansGrille**

Précondition : message ne contient que des caractères appartenant à l'ensemble {'A', ..., 'Z', '.', '.'}

Entrée : - grille : tableau 2D de alphabet\*alphabet 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 alphabet - 1 avec un pas de +1 Faire

Pour j de 0 à alphabet - 1 avec un pas de +1 Faire

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

Soit indice1 = i

Soit indice2 = j

Retourner // quitter la procédure

Fin Si

Fin Pour

Fin Pour

Fin

Fonction : **Decode2Gramme**

Précondition : message ne contient que des triplets caractères existant dans grille

Entrée : - grille : tableau 2D de alphabet\*alphabet chaînes de caractères

- message : chaîne de caractères

Local : - indice1, indice2 : entier

Sortie : - messageDecode : chaînes 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 = SousChaîne(message, i, i+3) // sous-chaîne de l'indice i à l'indice i+3

RechercherDansGrille(grille, msg, indice1, indice2) // modification de indice1 et indice2

Si indice1 ≠ -1 ET indice2 ≠ -1 Alors

Soit messageDecode = messageDecode + DonnerCaractere(indice1)

+ DonnerCaractere(indice2)

Fin Si

Fin Pour

Retourner messageDecode

Fin

Procédure : **Main**

Entrée : **void**

Local :  
- grille : **tableau 2D** de alphabet\*alphabet **chaînes de caractères**  
- alphabet : **entier**  
- messageU, messageUCode, messageUDecode : **chaîne de caractères**  
- ligneDepart1, colonneDepart1 : **entier**

Sortie : **void**

Début

Soit alphabet = 28 // { 'A', ... 'Z', ' ', '.' } = 26 + 1 + 1 = 28 caractères  
Soit ligneDepart1 = **SaisirEntier**()  
Soit colonneDepart1 = **SaisirEntier**()  
Soit grille = **CreerGrille**(alphabet, ligneDepart1, colonneDepart1)  
Soit messageU = **SaisirChaîne**()  
Soit messageUCode = **Code2Gramme**(grille, messageU)  
**Afficher**(messageUCode)  
Soit messageUDecode = **Decode2Gramme**(grille, messageUCode)  
**Afficher**(messageUDecode)

Fin