

## TD12 : éléments de correction

### Exercice 3

Fonction : **CalculVols**

Entrée : - liaisons : tableau 2D d'entiers

Local : - ligne, colonne : entiers

Sortie : - total : entier

Début

Soit total = 0

Pour ligne de 0 à liaisons.Hauteur() avec un pas de +1 Faire

| Pour colonne de 0 à liaisons.Largeur() avec un pas de +1 Faire

| | Soit total = total + liaisons[ligne, colonne]

| Fin Pour

Fin Pour

Retourner total

Fin

Fonction : **CalculVolsPilote**

Entrée : - liaisons : tableau 2D d'entiers

- pilote : entier

Local : - colonne : entier

Sortie : - total : entier

Début

Si pilote < 0 OU pilote ≥ liaisons.Hauteur() Alors

| Exception → indice de ligne invalide

Fin Si

Soit total = 0

Pour colonne de 0 à liaisons.Largeur() avec un pas de +1 Faire

| Soit total = total + liaisons[pilote, colonne]

Fin Pour

Retourner total

Fin

Fonction : **RecapAnnuel**

Entrée : - liaisons : tableau 2D d'entiers

- pilotes : liste de chaînes de caractères

Local : - indicePilote : entier

- pilote : chaîne de caractères

- vols : entier

Sortie : - recapitulatif : dictionnaire (clef : chaîne de caractères, valeur : entier)

Début

Soit recapitulatif = dictionnaire vide (clef : chaîne de caractères, valeur : entier)

Pour indicePilote de 0 à pilotes.Longueur() - 1 avec un pas de +1 Faire

| Soit pilote = pilotes[indicePilote]

| Soit vols = CalculVolsPilote(liaisons, indicePilote)

| recapitulatif.Ajouter(pilote, vols)

Fin Pour

Retourner recapitulatif

Fin

Procédure : **AfficherRecapitulatif**

Entrée : - recap : dictionnaire (clef : chaîne de caractères, valeur : entier)

Local : /

Sortie : void

Début

Pour Chaque paire de clef, valeur dans recap Faire

| Afficher("{clef} : {valeur}\n")

Fin Pour Chaque

Fin

**Fonction : DestinationsPilotes**

**Entrée :**

- liaisons : tableau 2D d'entiers
- pilotes : liste de chaînes de caractères
- destinations : liste de chaînes de caractères

**Local :**

- indicePilote, indiceDestination : entiers
- pilotesDestination : liste de chaînes de caractères
- pilote : chaîne de caractères

**Sortie :**

- destPil : dictionnaire (clef : chaîne de caractères, valeur : liste de chaînes de caractères)

**Début**

```
Soit destPil = dictionnaire vide (clef : chaîne de caractères, valeur : liste de chaînes de caractères)
Pour indiceDestination de 0 à destinations.Longueur() - 1 avec un pas de +1 Faire
    Soit destination = destinations[indiceDestination]
    Soit pilotesDestination = liste de chaînes de caractères vide
    Pour indicePilote de 0 à pilotes.Longueur() - 1 avec un pas de +1 Faire
        Si liaisons[indicePilote, indiceDestination] > 0 Alors
            pilotesDestinations.Ajouter(pilotes[indicePilote])
        Fin Si
    Fin Pour
    destPil.Ajouter(destination, pilotesDestination)
Fin Pour
Retourner destPil
```

**Fin**

**Procédure : AfficherDestinationsPilotes**

**Entrée :**

- destPil : dictionnaire (clef : chaîne de caractères, valeur : liste de chaînes de caractères)

**Local :**

- kvp : couple de clef, valeur de destPil

**Sortie :**

- void

**Début**

```
Pour Chaque paire de clef, valeur dans destPil Faire
    Afficher("{clef} : ")
    Pour Chaque destination dans valeur Faire
        Afficher("{destination} ")
    Fin Pour Chaque
    Afficher("\n")
Fin Pour Chaque
```

**Fin**

**Procédure : AjouterVol (bonus)**

**Entrée :**

- liaisons : tableau 2D d'entiers
- pilotes : liste de chaînes de caractères
- destinations : liste de chaînes de caractères
- pilote : chaîne de caractères
- destination : chaîne de caractères

**Local :**

- indicePilote, indiceDestination : entiers

**Sortie :**

- void

**Début**

```
Soit indicePilote = pilotes.PremiereOccurrences(pilote)
Soit indiceDestination = destinations.PremiereOccurrences(destination)
Si indicePilote ≥ 0 ET indiceDestination ≥ 0 Alors
    Soit liaisons[indicePilote, indiceDestination] = liaisons[indicePilote, indiceDestination] + 1
Fin Si
```

**Fin**

Fonction : **AjouterPilote** (bonus)

Entrée :  
- liaisons : tableau 2D d'entiers  
- pilotes : liste de chaînes de caractères  
- pilote : chaîne de caractères  
Local :  
- ligne, colonne : entiers  
Sortie :  
- liaisonsEtendues : tableau 2D d'entiers

Début

```
|      Si pilotes.Contient(pilote) == FAUX Alors
|          Soit liaisonsEtendues = tableau 2D de liaisons.Hauteur() + 1 * liaisons.Largeur() entiers
|          Pour ligne de 0 à liaisons.Hauteur() - 1 avec un pas de +1 Faire
|              Pour colonne de 0 à liaisons.Largeur() - 1 avec un pas de +1 Faire
|                  Soit liaisonsEtendues[ligne, colonne] = liaisons[ligne, colonne]
|              Fin Pour
|          Fin Pour
|      Fin Si
|      Retourner liaisonsEtendues
Fin
```

Fonction : **AjouterDestination** (bonus)

Entrée :  
- liaisons : tableau 2D de d'entiers  
- destinations : liste de chaînes de caractères  
- destination : chaîne de caractères  
Local :  
- indicePilote, indiceDestination : entiers  
Sortie :  
- liaisonsEtendues : tableau 2D d'entiers

Début

```
|      Si destinations.Contient(destination) == FAUX Alors
|          Soit liaisonsEtendues = tableau 2D de liaisons.Hauteur() * liaisons.Largeur() + 1 entiers
|          Pour ligne de 0 à liaisons.Hauteur() - 1 avec un pas de +1 Faire
|              Pour colonne de 0 à liaisons.Largeur() - 1 avec un pas de +1 Faire
|                  Soit liaisonsEtendues[ligne, colonne] = liaisons[ligne, colonne]
|              Fin Pour
|          Fin Pour
|      Fin Si
|      Retourner liaisonsEtendues
Fin
```