

Contrôle de novembre : éléments de correction et barème

Exercice 1 (1 point)

Soit EDT = tableau 2D de 11*6 chaînes de caractères (1 point)

Exercice 2 (14 points)

Fonction : **TotalParUE** (3 points)

Entrée :
- EDT : tableau 2D de 11*6 chaînes de caractères
- UE : chaîne de caractères // nom de l'UE à rechercher

Local :
- ligne, colonne : entiers

Sortie :
- total : entier

Début

Soit total = 0

Pour ligne de 1 à 10 avec un pas de +1 Faire

 Pour colonne de 1 à 5 avec un pas de +1 Faire

 Si Comparer(EDT[ligne, colonne], UE) est VRAI Alors

 Soit total = total + 1

 Fin Si

 Fin Pour

Fin Pour

Retourner total

Fin

Fonction : **NumJour** (1 point)

Entrée :
- nomJour : chaîne de caractères // nom du jour de la semaine

Local :
/

Sortie :
- colonne : entier

Début

Soit colonne = -1

Si Comparer(nomJour, "Lundi") est VRAI Alors

 Soit colonne = 1

Sinon Si Comparer(nomJour, "Mardi") est VRAI Alors

 Soit colonne = 2

Sinon Si Comparer(nomJour, "Mercredi") est VRAI Alors

 Soit colonne = 3

Sinon Si Comparer(nomJour, "Jeudi") est VRAI Alors

 Soit colonne = 4

Sinon Si Comparer(nomJour, "Vendredi") est VRAI Alors

 Soit colonne = 5

Fin Si

Retourner colonne

Fin

Remarque : on pouvait aussi utiliser le tableau EDT. Il suffit de parcourir la première ligne (indices de colonne de 1 à 5) et si Comparer(EDT[0, colonne], nomJour) est vrai, alors retourner l'indice de colonne.

Fonction : **TestTropDheure** (5 points)

Précondition : UE est une chaîne correspondant à un nom d'UE valide

Entrée :

- EDT : **tableau 2D** de 11*6 **chaînes de caractères**
- UE : **chaîne de caractères**
- jour : **chaîne de caractères**

Local :

- colonne : **entier**
- heures : **entier**
- ligne : **entier**

Sortie :

- **booléen**

Début

Soit colonne = **NumJour**(jour)

Soit heures = 0

Pour ligne **de** 1 **à** 10 **avec un pas de** +1 **Faire**

Si **Comparer**(EDT[ligne, colonne], UE) est VRAI **Alors**

 Soit heures = heures + 1

Fin Si

Si heures > 3 **Alors**

Retourner VRAI

Fin Si

Fin Pour

Retourner FAUX

Fin

Procédure : **Main** (5 points)

Principe : Parcourir un tableau contenant le nom toutes les UE. Pour chacune, appeler les fonctions TestTropDheure (pour chaque jour de la semaine) avec comme paramètre le retour de la fonction NumJour et totalParUE puis afficher les résultats dans la console.

Entrée :

- **void**

Local :

- EDT : **tableau 2D** de 11*6 **chaînes de caractères**
- tabUE : **tableau 1D** de 7 **chaînes de caractères**
- jours : **tableau 1D** de 5 **chaînes de caractères**
- numJour : **entier**
- UE : **chaîne de caractères**
- nbHeures : **entier**
- tropDheures : **booléen**

Sortie :

- **void**

Début

Soit tabUE = **tableau 1D** de 7 **chaînes de caractères** initialisé avec

 {"Algorithmique", "Math pour l'info", "Archi/Système", "Réseaux", "Programmation C",
 "Conduite projet", "Langue"}

Soit jours = **tableau 1D** de 5 **chaînes de caractères** initialisé avec

 {"Lundi", "Mardi", "Mercredi", "Jeudi", "Vendredi"}

Pour i **de** 0 **à** 6 **avec un pas de** +1 **Faire**

 Soit UE = tabUE[i]

 Soit nbHeures = **TotalParUE**(EDT, UE)

 Soit tropDheures = FAUX

Pour j **de** 0 **à** 4 **avec un pas de** +1 **Faire**

 Soit tropDheure = tropDheure OU **TestTropDheure**(EDT, jour)

Fin Pour

Si tropDheure est VRAI **Alors**

 Afficher("{UE} : {nbHeures} (dépassement)")

Sinon

 Afficher("{UE} : {nbHeures}")

Fin Si

Fin Pour

Fin

Exercice 3 (5 points)

Soit **structure StructUE** (1 point)

```
{  
    Champ nom : chaîne de caractères  
    Champ nbHeures : entier  
}
```

Soit tabRecapUE = **tableau 1D** de 7 **structure StructUE** (1 point)

Procédure : InitUE (3 points)

Entrée :

- tabRecapUE : **tableau** de 7 structure StructUE (modifié)
- EDT : **tableau 2D** de 11*6 chaînes de caractères initialisé avec
{"Algorithmique", "Math pour l'info", "Archi/Système", "Réseaux", "Programmation C",
"Conduite projet", "Langue"}
- tabUE : **tableau 1D** de 7 chaînes de caractères

Local :

- heures : **entier**
- ligne : **entier**

Sortie : void

Début

```
| Pour i de 0 à 6 avec un pas de +1 Faire  
| | Soit tabRecapUE[i].nom = tabUE[i]  
| | Soit tabRecapUE[i].nbHeures = TotalParUE(EDT, tabUE[i])  
| Fin Pour
```

Fin