

Consignes :

- Vous devez répondre à chaque question en donnant l'algorithme correspondant. Sauf indication contraire, vous n'avez pas à donner l'idée/principe de vos algorithmes
- Vous n'avez pas à fournir de code C#.

Dans la suite, on suppose connues les fonctions

- **compare** qui renvoie Vrai si les deux chaînes de caractères (*string*) passées en argument sont identiques. Faux sinon.
- **copy** qui copie une chaîne de caractères (*string*) S1 passée en premier argument dans une chaîne S2 passée en 2ème argument

Ces deux fonctions n'ont pas de précondition

L'emploi du temps d'un étudiant est stocké sous la forme d'un tableau à deux dimensions (11 lignes, 6 colonnes), comme illustré ci-dessous.

	Lundi	Mardi	Mercredi	Jeudi	Vendredi
8h-9h	Math pour l'info	Archi/Système	Algorithmique	Langue	Programmation C
9h-10h	Math pour l'info	Archi/Système	Algorithmique	Langue	Programmation C
10h-11h	Programmation C	Programmation C	Archi/Système	Archi/Système	Conduite projet
11h-12h	Programmation C	Programmation C	Archi/Système	Archi/Système	Conduite projet
12h-13h	Libre	Libre	Libre	Libre	Libre
13h-14h	Libre	Libre	Libre	Archi/Système	Langue
14h-15h	Langue	Programmation C	Libre	Conduite projet	Math pour l'info
15h-16h	Algorithmique	Programmation C	Programmation C	Algorithmique	Math pour l'info
16h-17h	Algorithmique	Algorithmique	Programmation C	Algorithmique	Archi/Système
17h-18h	Libre	Algorithmique	Libre	Libre	Archi/Système

- Ce tableau sera appelé EDT dans la suite. Pour pouvoir être utilisé par une fonction, il devra être passé en argument.
- Chaque élément de ce tableau est représenté par une chaîne de caractères du type *string*. On suppose qu'il n'y a pas de faute d'orthographe. Une heure sans cours est notée "Libre".
- Chaque UE (Unité d'Enseignement) est donnée par un nom unique. On suppose qu'il y a 7 UE différentes données par :
TabUE : tableau de 7 chaînes de caractères, initialisé à {"Algorithmique", "Math pour l'info", "Archi/Système", "Réseaux", "Programmation C", "Conduite projet", "Langue"}

Exercice 1 (1,5 point) – Création de l'emploi du temps

Donner la déclaration de cet emploi du temps

// On ne vous demande pas d'écrire l'algorithme d'initialisation.

Exercice 2 (10,5 points) – Analyse d'un emploi du temps

1. (2 points) Écrire la fonction **totalParUE** qui renvoie le nombre d'heures de cours de la semaine pour une UE donnée en argument. Elle rendra 0 si cette UE n'apparaît pas dans l'emploi du temps de la semaine.
2. (3,5 points) Écrire une fonction **testTropDheures** qui teste si pour une UE et un jour de la semaine donnés en arguments, il y a plus de 3 heures de cours de cet UE dans cette journée. Elle rendra Vrai si c'est le cas. Faux sinon.
3. (5 points) Écrire le **Main** qui affiche pour chaque UE, son nom et son nombre d'heures de cours dans l'emploi du temps. Il affiche aussi une alarme si cette UE a plus de 3h dans une journée. Enfin, il affiche le total d'heures de cours.

On vous demande de préciser l'idée/principe de l'algorithme.

Exemple d'exécution

Algorithmique : 4

Math pour l'info : 4

Archi/Système : 9

Réseaux : 0

Programmation C : 10 (Dépassement)

Conduite projet : 1

Langue : 5

Total : 33

Exercice 2 (8 points) – Gestion des UE

On désire gérer un tableau TabRecapUE où pour chaque UE, on trouvera les informations suivantes :

- Nom de l'UE
- Nombre d'heures dans l'emploi du temps

1. (1,5 point) Définir une structure de données permettant de représenter une UE
2. (1,5 point) Déclarer le tableau TabRecapUE
3. (5 points) Écrire une procédure **InitUE** qui remplit le nom de chaque UE du tableau TabRecapUE et le nombre d'heures dans l'emploi du temps EDT