


Institut Supérieur d'Informatique et de Mathématiques 	Année Universitaire : 2023-2024
	<b>Examen (Session Principale)</b> Matière : <b>Prog. temps réel et concurrente</b> Filière : <b>MP2-GL</b> Enseignant : <b>Taoufik Sakka Rouis</b>

### Exercice 1 : (3 Points)

En termes de concurrence, Ada 95 introduit un nouveau concept par rapport à Ada 83. Pouvez-vous expliquer les avantages de l'utilisation de ce nouveau concept ?

### Exercice 2 : (3 +6 Points)

Supposant qu'on a un programme Ada constitué de deux tâches : tic et tac.

Chaque tâche utilise un compteur commun (variable globale à la procédure principale de type entière) et effectue un certain nombre de fois l'un des traitements suivants :

Pour la tâche tic :

- Incrémentation du compteur.
- Affichage de la valeur du compteur et du mot "tic"

Pour la tâche tac :

- Décrémentation du compteur.
- Affichage de la valeur du compteur et du mot "tac"

La procédure principale contient une seule instruction qui consiste à initialiser le compteur à zéro.

### Partie 1 :

- Combien de processus ce programme contient-il ?
- Quand les tâches commencent-elles leurs exécutions ?
- Quand la procédure principale se termine-t-elle ?

### Partie 2 :

Maintenant, pour assurer le bon usage du compteur commun par les deux tâches au même temps, on vous propose d'ajouter une tâche serveuse qui permet de protéger ce compteur.

On suppose que : Le traitement de la tâche 1 (incrément et affichage) se répète 10 fois

Le traitement de la tâche 2 (décrément et affichage) se répète 10 fois

Proposer une implémentation pour ce programme.

### Exercice 3 : (8 Points)

Imaginant un programme simulant le problème des lecteurs/rédacteurs dans un environnement partagé. Vous disposerez d'une tâche serveuse représentant une ressource partagée (par exemple, une base de données, une valeur entière, etc.), ainsi que de deux types de tâches actrices : les lecteurs et les rédacteurs.

1. Complétez la tâche serveuse qui permet de représenter la ressource partagée (par exemple, un simple entier) avec une procédure « Lire » pour les lecteurs et une procédure « Ecrire » pour les rédacteurs. Assurez-vous que les rédacteurs et les lecteurs ont un accès exclusif à la ressource. (N'écrivez que le body du Serveur)
2. Complétez la tâche actrice « Rédacteur » qui écrit dans la ressource en utilisant la procédure « Ecrire ». (N'écrivez que le body du Redacteur)

<pre>with Ada.Text_IO; use Ada.Text_IO; with Ada.Integer_Text_IO; use Ada.Integer_Text_IO; procedure Lecteurs_Redacteurs is    task type Serveur is     entry Lire (Contenu : out Integer);     entry Ecrire (Nouveau_Contenu : in Integer);   end Serveur;    task type Lecteur (Id : Integer);   task type Redacteur (Id : Integer);    task body Serveur is     Contenu_Partage : Integer := 0;   begin     ..... - <b>Partie 1</b>   end Serveur;</pre>	<pre>task body Lecteur is   Contenu_Lu : Integer; begin   loop     delay 1.0; -- Temps entre les lectures     Serveur.Lire(Contenu_Lu);     Put_Line("Lecteur    numéro    " &amp; Integer'Image(Id) &amp; " lit la valeur : " &amp; Integer'Image(Contenu_Lu));   end loop; end Lecteur;  task body Redacteur is   ..... - <b>Partie 2</b>  end Redacteur;  Serveur_Instance : Serveur; Lecteur1 : Lecteur (1); Lecteur2 : Lecteur (2); Redacteur1 : Redacteur (1); Redacteur2 : Redacteur (2);  begin   null; end Lecteurs_Redacteurs;</pre>
---	--