

**Contrôle INF346 - 14 Mai 2008**  
**Sans documents (dictionnaires autorisés)**  
**Durée : 1 heure 30**

Si vous n'êtes pas francophone, vous pouvez répondre aux questions en anglais.

**Merci de répondre à chaque partie sur une feuille séparée en indiquant bien votre nom sur chaque feuille.**

**1<sup>ère</sup> Partie (4 points).**

**I.1. Communication de groupe (2 points)**

**Rappeler le fonctionnement de la diffusion par horloge vectorielle et justifier les deux règles de validation d'un message avant transmission à l'application.**

**I.1. Réponse :**

On maintient des horloges de Mattern. Chaque message est diffusé à chacun des membres du groupe et contient l'horloge vectorielle de l'émetteur. (1 point)

Un message d'horloge  $V_m$  en provenance de  $i$  reçu par  $j$  n'est délivré auprès de l'application que si (0.5 point par règle) :

1/  $V_j(i) = V_m(i) - 1$

Justification : toutes les diffusions précédentes de  $i$  ont été reçues par  $j$

2/ qq soit  $k \neq i$   $V_m(k) \leq V_j(k)$

Justification : toutes les diffusions reçues par  $i$  ont été reçues par  $j$ .

**I.2. Modèle de répartition en Ada (2 points)**

**Expliquer comment en Ada un sous-programme P peut être rendu asynchrone ?**

**Quelle propriété doit-il respecter ?**

**Donner un exemple d'application répartie utilisant 2 sous-programmes appelables à distance dont le comportement devient non-déterministe lorsque ces 2 sous-programmes sont rendus asynchrones.**

**I.2. Réponse :**

pragma Asynchronous (P); (0.5 point)

P ne doit pas avoir de paramètres out, inout ou de valeur de retour (donc une procédure pas une fonction). Toute exception est ignorée. (0.5 point)

P1 (X) se trouve sur le nœud N3 et fait  $N=X$ . N est une variable local du nœud N3.  
P2 (X) se trouve sur le nœud N2 et appelle P1 (2\*X) donc sur N3.

Sur le nœud N1 on appelle P2(1) puis P1(1) (0.5 point pour la description de l'exemple du cours)

En synchrone, N vaut toujours 1 (P1(1) est toujours exécuté en dernier)

En asynchrone, l'appel sur N2 de P1 peut être retardé (latence réseau) et l'appel à partir de N1 de P1 peut se terminer avant que P1(2) à partir de N2 ne soit fait. Donc N peut valoir 2. (0.5 point pour le scénario aboutissant au non déterminisme).

## 2ème Partie (6 points)

**Merci de répondre à chaque partie sur une feuille séparée en indiquant bien votre nom sur chaque feuille.**

### II.1. Composants

**a) Le modèle de programmation composant offre plusieurs avancées comparés au modèle de programmation orienté objets. Ces avantages permettent de répondre à plusieurs limitations du modèle objet répartis proposés par CORBA. En vous basant sur CCM, indiquez trois de ces limitations, que vous justifierez. ( 1.5 points )**

**b) En les détaillant, comparez les étapes de construction d'une application CORBA et d'une application CORBA CCM. Vous indiquerez l'apport de CCM. ( 1.5 points )**

### II.2 Intergiciels Orientés Messages

On souhaite concevoir une application répartie permettant à un groupe de personnes d'échanger des informations. L'objectif est que chaque personne dispose d'un dispositif embarqué comportant plusieurs informations nominatives et professionnelles. Cette "carte de visite virtuelle" est transmise à toute personne se trouvant à proximité. Les informations sont localisées, et ne doivent pas être transmises à des personnes en dehors d'un périmètre donné (salle de réunion par exemple). On suppose qu'un dispositif externe fournit cette information de localisation.

**a) En le comparant à d'autres modèles de répartition (peer to peer, RPC, objets répartis), justifiez le choix d'un intergiciel orienté messages pour remplir ce rôle. ( 1 point )**

b) On se place dans le cadre d'une architecture JMS. Chaque utilisateur dispose d'une file de message qui lui est propre pour recevoir les "cartes de visite virtuelles" d'autres participants. De plus, on suppose qu'il existe un canal de communication (topic) "salle de réunions" auquel chaque dispositif s'abonne lors de l'entrée dans la salle. Cet événement est transmis via le mécanisme de localisation.

Périodiquement, chaque dispositif envoie les informations concernant son porteur. Parallèlement, il reçoit les messages des autres participants, les traite en les stockant dans une table. On souhaite éviter tout mécanisme d'attente active et ne pas stocker inutilement de messages dans le provider.

**Proposez un schéma d'implantation de ce scénario: vous listerez les entités JMS et Java que vous comptez utiliser, et indiquerez succinctement comment vous les utilisez. ( 1 point )**

**c) On se place dans le cadre d'une architecture JMS. En rappelant son rôle, proposez une architecture pour le "provider" JMS qui permette de supporter un mode fortement déconnecté, réparti sur plusieurs sites. Ceci doit permettre à l'application de fonctionner sur un site géographiquement éclaté (par exemples les différentes salles d'un hôtel) ( 1 point )**

**3ème Partie. Mémoire Partagée Répartie (4 points)**

**Merci de répondre à chaque partie sur une feuille séparée en indiquant bien votre nom sur chaque feuille.**

On rappelle la notation introduite en cours :  $W_i(x)a$  et  $R_i(x)b$  veulent dire que le processus  $P_i$  écrit la valeur  $a$  dans la variable  $x$  et que la lecture de la variable  $x$  par le processus  $P_i$  retourne la valeur  $b$ .

**III.1 Rappeler la définition de la cohérence séquentielle (1 point)**

**III.2 Observez les deux séquences suivantes et indiquez si chacune d'elle respecte la cohérence séquentielle et pourquoi (1 point)**

P1	$W1(x)a$		
P2		$W2(x)b$	
P3		$R3(x)b$	$R3(x)a$
P4		$R4(x)b$	$R4(x)a$

P1	$W1(x)a$		
P2		$W2(x)b$	
P3		$R3(x)b$	$R3(x)a$
P4		$R4(x)a$	$R4(x)b$

**III.3 Cohérence causale.**

On rappelle l'ordre causal défini par Lamport. Soient  $Opé1$  et  $Opé2$ , deux opérations (lecture / écriture) ;  $Opé1$  précède  $Opé2$  causalement si :

- 1/  $Opé1$  précède  $Opé2$  sur le même processeur, ou
- 2/  $Opé2$  est une lecture de la valeur écrite par  $Opé1$ , ou
- 3/ il existe  $Opé3$  telle que  $Opé1$  précède  $Opé3$  causalement et  $Opé3$  précède  $Opé2$  causalement.

$Opé1$  et  $Opé2$  sont concurrentes elles ne sont pas en dépendance causale.

On rappelle que la cohérence causale respecte l'ordre causal pour les opérations causalement dépendantes mais deux écritures concurrentes peuvent être vues dans des ordres différents sur des processeurs différents.

**La séquence suivante respecte-t-elle la cohérence séquentielle ? La cohérence causale ? (2 points)**

P1	$W1(x)a$		$W1(x)c$	
P2		$R2(x)a$	$W2(x)b$	
P3		$R3(x)a$		$R3(x)c$ $R3(x)b$
P4		$R4(x)a$		$R4(x)b$ $R4(x)c$

### **Correction**

III.1. Un ensemble d'opérations respecte la cohérence séquentielle si toutes les opérations sont vues par tous les processeurs dans un ordre correspondant à une exécution séquentielle, et si sur chaque processeur l'ordre correspond à l'ordre d'exécution du programme.

III.2. La première séquence respecte la cohérence séquentielle, la seconde pas car P3 et P4 ne voient pas la même séquence d'exécution.

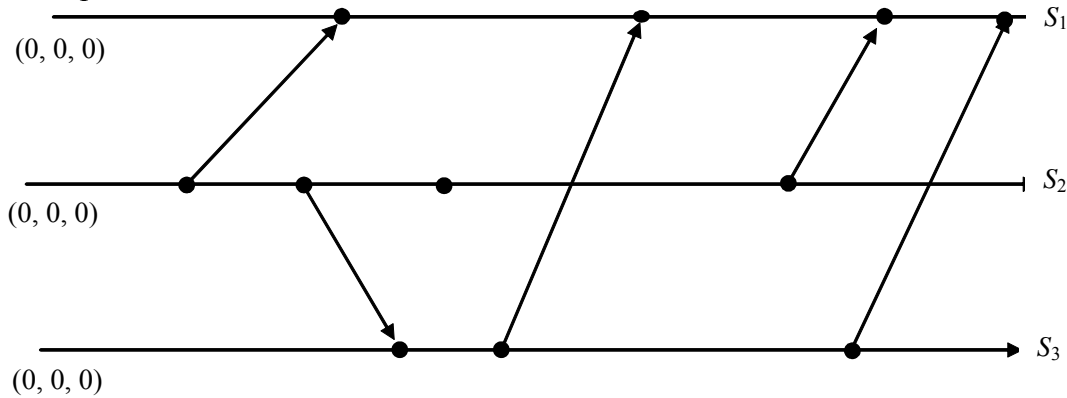
III.3 La séquence ne respecte pas l'ordre séquentiel car P3 et P4 ne voient pas la même séquence d'exécution. Elle respecte la cohérence causale car il n'y a pas de dépendance causale entre  $W2(x)_b$  et  $W1(x)_c$  qui sont donc concurrentes et rien ne s'oppose donc à ce que P3 et P4 les voient dans un ordre différent.

Nom :

Prénom :

**4ème Partie - Questionnaires d'algorithmique du contrôle réparti à rendre avec la copie (6 points)**

**Question 1 :** Le dessin ci-dessous représente le déroulement du temps sur trois sites ; chaque ligne horizontale correspond à un site, le temps s'écoule de la gauche vers la droite. Chaque point noir correspond à un événement. Chaque flèche correspond à un message envoyé d'un site à l'autre. Les sites utilisent des horloges vectorielles. Indiquer à côté de chaque événement la date de celui-ci et à côté de chaque flèche l'estampille du message correspondant.



**Question 2**

On considère un système distribué dans lequel les sites (les processus) doivent effectuer des transactions nécessitant l'usage simultané de plusieurs ressources partagées ; chaque ressource est utilisée en exclusion mutuelle. Indiquer le principe d'un algorithme, au choix, permettant de prévenir un interblocage du système consécutif à ces transactions (le cadre, qui continue à la page suivante, permet une réponse longue, mais cela n'exclut pas une réponse en quelques lignes).

**Réponse de la question 2 :**

**Nom :**

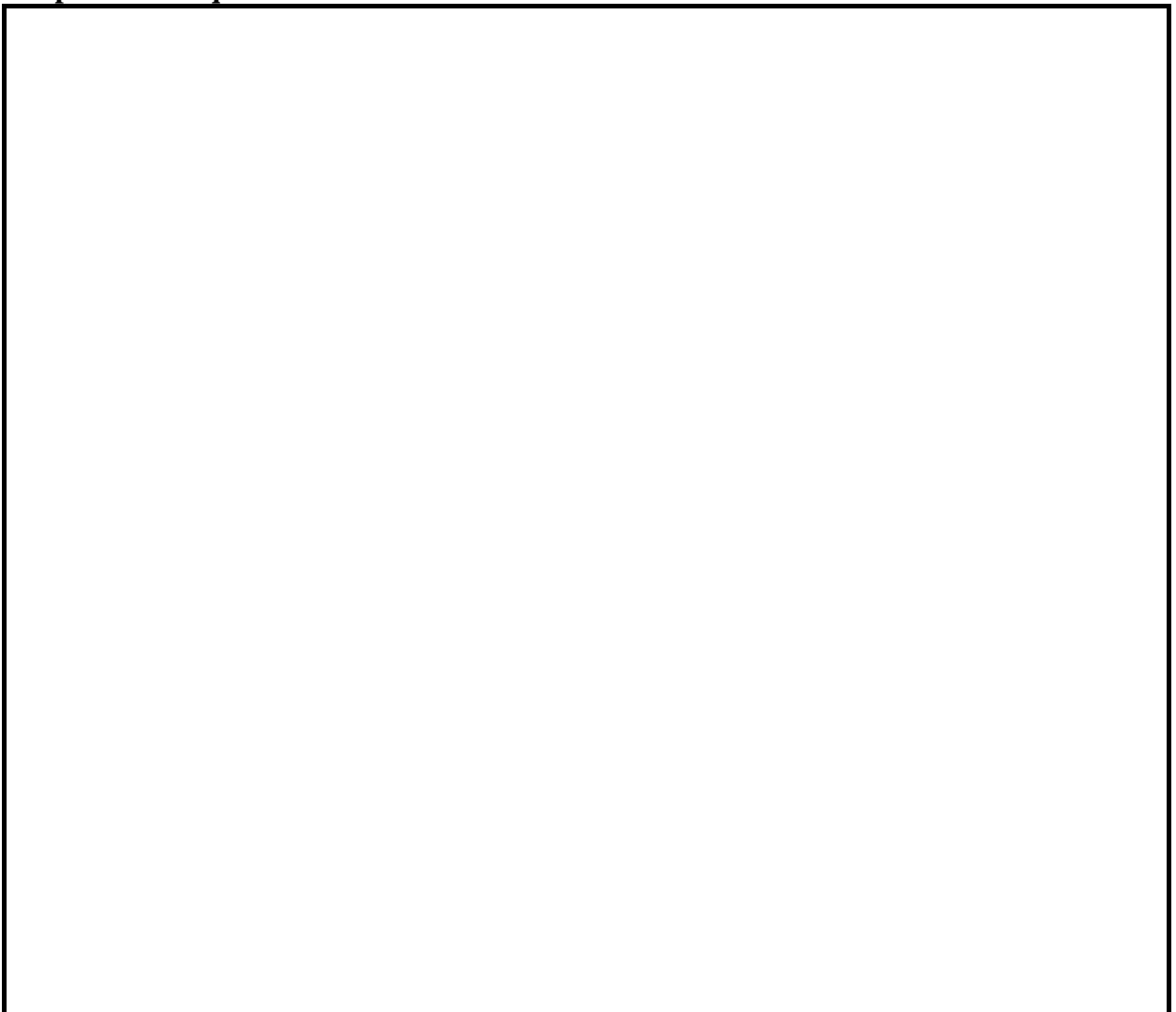
**Prénom :**



**Question 3**

Définir ce qu'est un état global cohérent d'un système distribué.

**Réponse de la question 3 :**

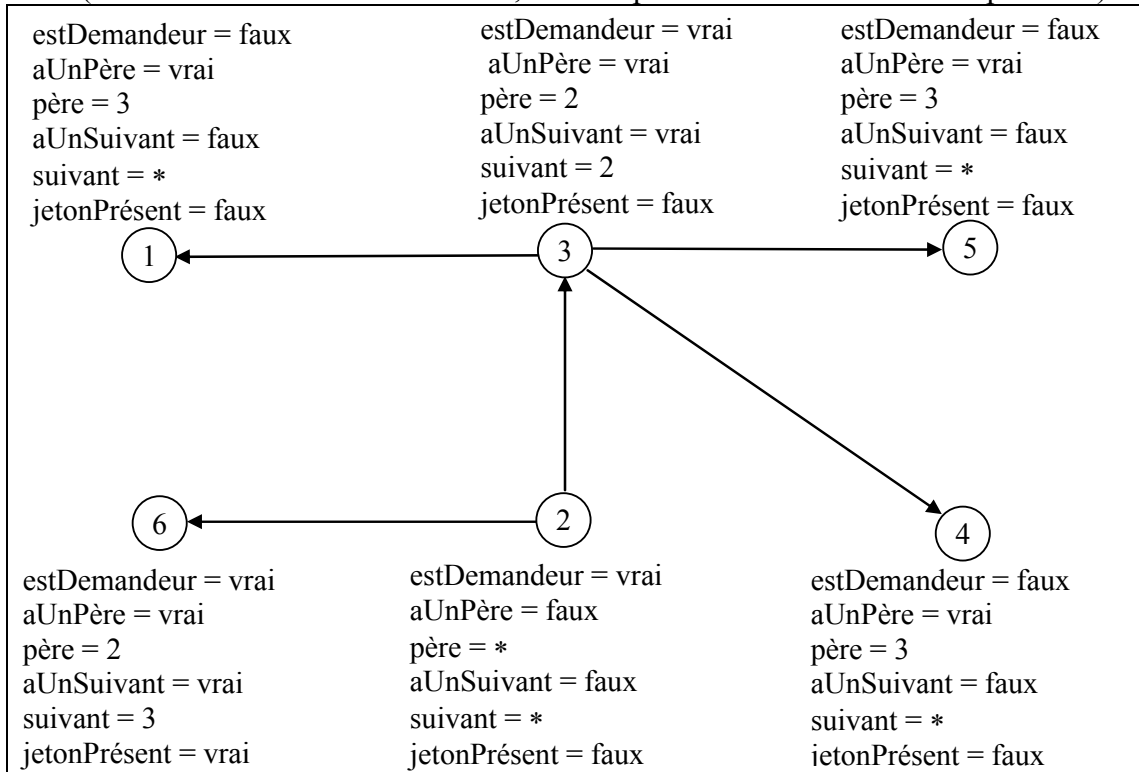


Nom :

Prénom :

**Question 4 :**

On utilise l'algorithme de Naimi et Tréhel (programmé en TP) pour l'utilisation d'une même ressource en exclusion mutuelle ; on suppose que l'arborescence est dans l'état représenté ci-dessous (avec les variables utilisées en TP, une \* représente une valeur sans importance) :



On suppose que les sites 4 et 5 demandent successivement la ressource, et que la demande du site 4 arrive au site 3 avant celle du site 5. Le jeton, pendant ce temps, reste chez le site 6. Dessiner l'arborescence consécutive à ces deux demandes en précisant les valeurs des différentes variables significatives non précisées.

Nom :

Prénom :

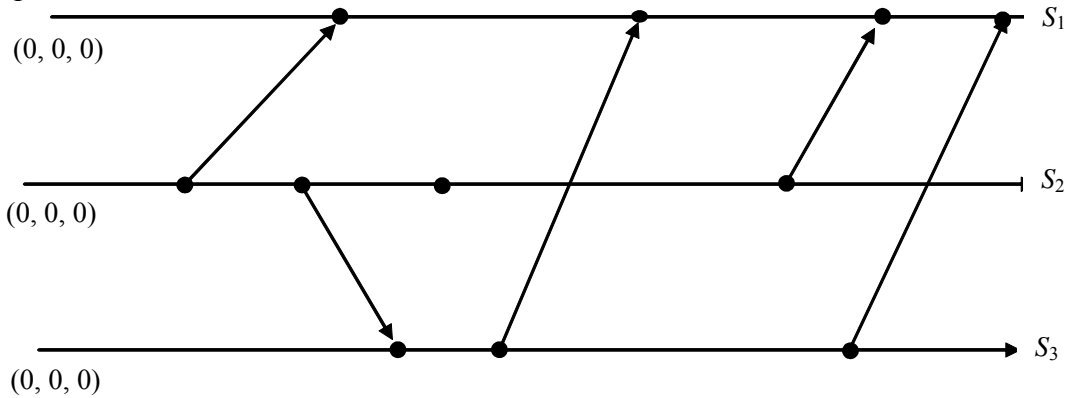
<p>estDemandeur = faux aUnPère = père = aUnSuivant = faux suivant = * jetonPrésent = faux</p> <p>①</p>	<p>estDemandeur = aUnPère = père = aUnSuivant = vrai suivant = 2 jetonPrésent = faux</p> <p>③</p>	<p>estDemandeur = aUnPère = père = aUnSuivant = suivant = jetonPrésent = faux</p> <p>⑤</p>
<p>⑥</p> <p>estDemandeur = vrai aUnPère = père = aUnSuivant = vrai suivant = 3 jetonPrésent = vrai</p>	<p>②</p> <p>estDemandeur = aUnPère = père = aUnSuivant = suivant = jetonPrésent = faux</p>	<p>④</p> <p>estDemandeur = aUnPère = père = aUnSuivant = suivant = jetonPrésent = faux</p>

Nom :

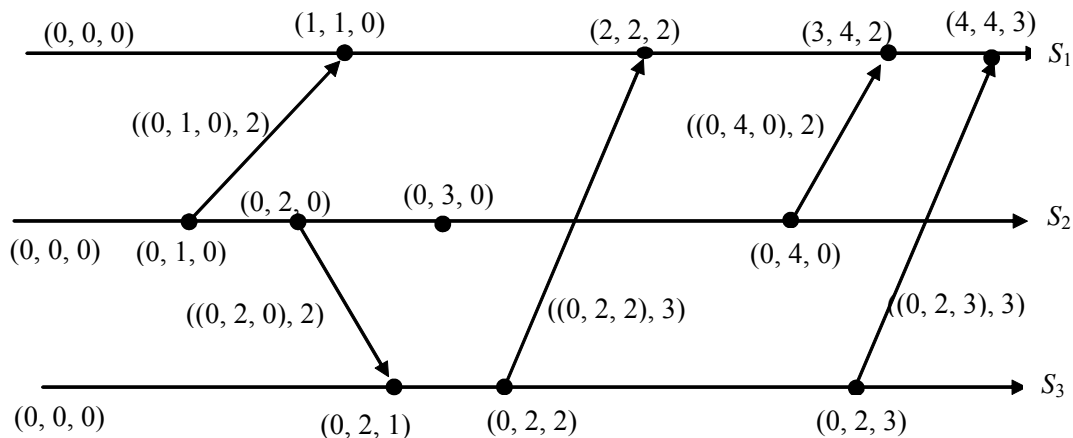
Prénom :

**Questions d'algorithmique du contrôle réparti 08**  
**Corrigé**

**Question 1 :** Le dessin ci-dessous représente le déroulement du temps sur trois sites ; chaque ligne horizontale correspond à un site, le temps s'écoule de la gauche vers la droite. Chaque point noir correspond à un événement. Chaque flèche correspond à un message envoyé d'un site à l'autre. Les sites utilisent des horloges vectorielles. Indiquer à côté de chaque événement la date de celui-ci et à côté de chaque flèche l'estampille du message correspondant.



**Réponse**



**Question 2**

On considère un système distribué dans lequel les sites (les processus) doivent effectuer des transactions nécessitant l'usage simultané de plusieurs ressources partagées ; chaque ressource est utilisée en exclusion mutuelle. Indiquer le principe d'un algorithme, au choix, permettant de prévenir un interblocage du système consécutif à ces transactions.

**Réponse**

Voir le polycopié.

**Question 3**

Définir ce qu'est un état global cohérent d'un système distribué.

**Réponse**

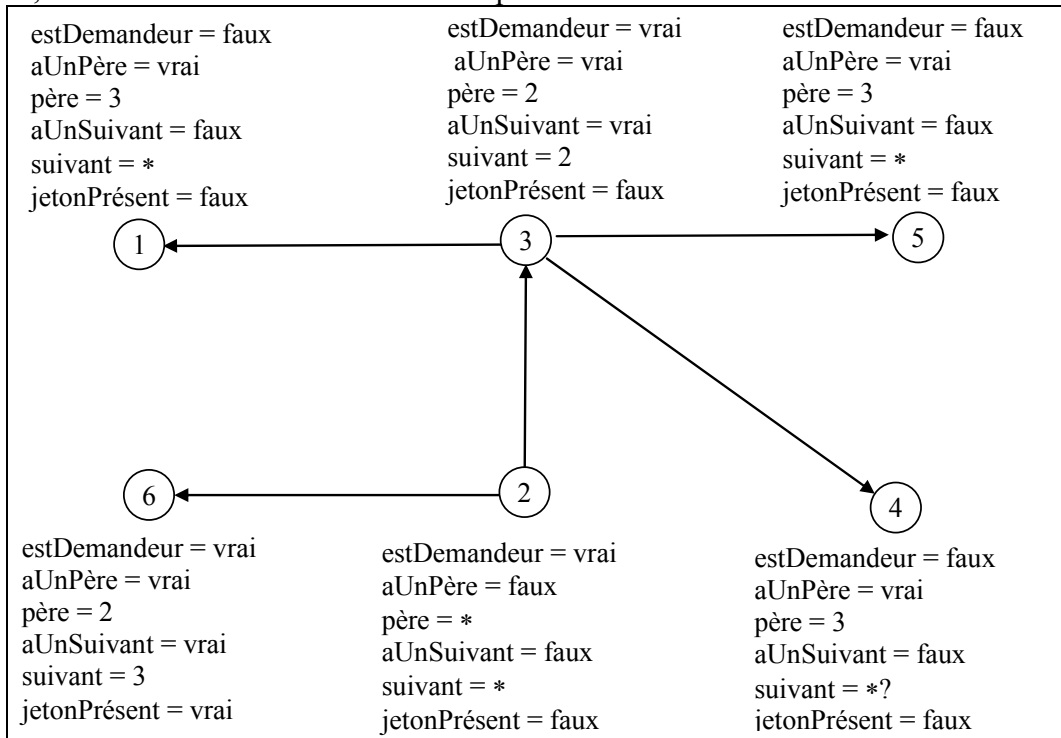
Voir le polycopié

Nom :

Prénom :

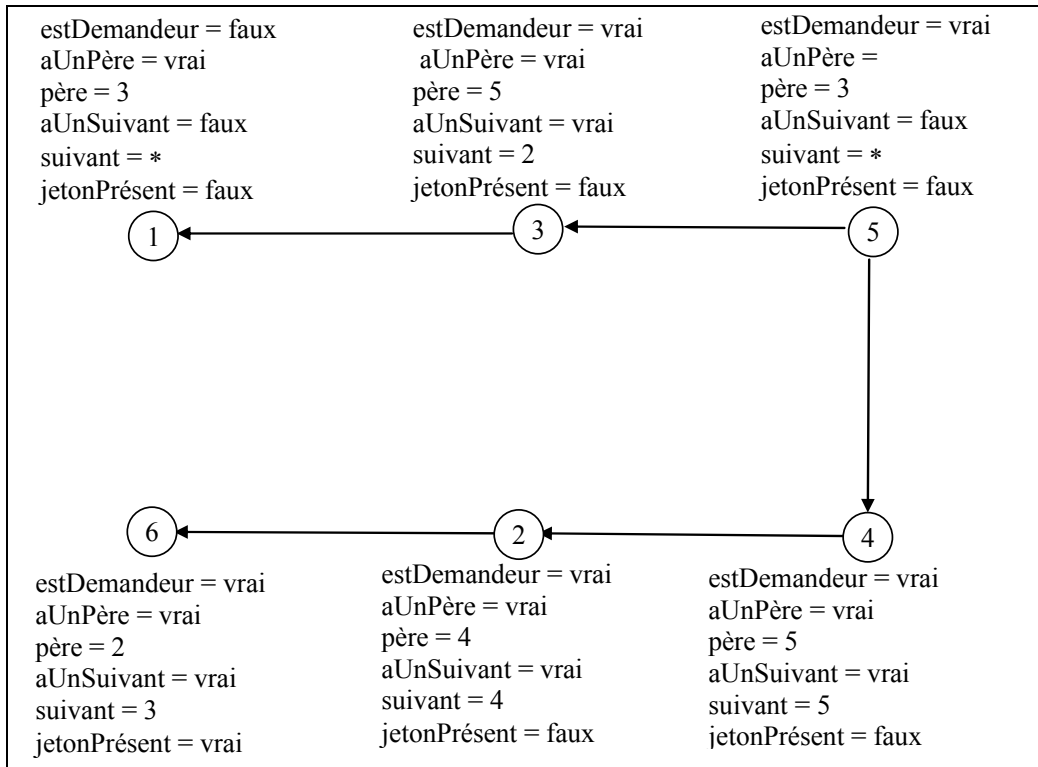
**Question 4 :**

On utilise l'algorithme de Naimi et Tréhel (programmé en TP) pour l'utilisation d'une même ressource en exclusion mutuelle ; on suppose que l'arborescence est dans l'état représenté ci-dessous, les valeurs des variables étant indiquées à côté des sites :



On suppose que les sites 4 et 5 demandent la ressource, presque en même temps, et que la demande du site 4 arrive au site 3 avant celle du site 5. Le jeton, pendant ce temps, reste chez le site 6. Dessiner l'arborescence consécutive à ces deux demandes en complétant les valeurs des différentes variables.

**Réponse**



**Nom :**

**Prénom :**