

UNIVERSITÉ DE SHERBROOKE
DÉPARTEMENT D'INFORMATIQUE

IFT 630

Processus concurrents et parallélisme

Examen périodique

Le mardi 22 février 2021
de 9 h à 9 h 50

Professeur : Gabriel Girard

- Notes :
- Toute documentation est permise.
 - Répondez dans les espaces prévus à cet effet.
 - Cet examen comporte trois questions sur huit pages.
 - Le total de l'examen est sur 100.
 - **Justifiez chacune de vos réponses**.

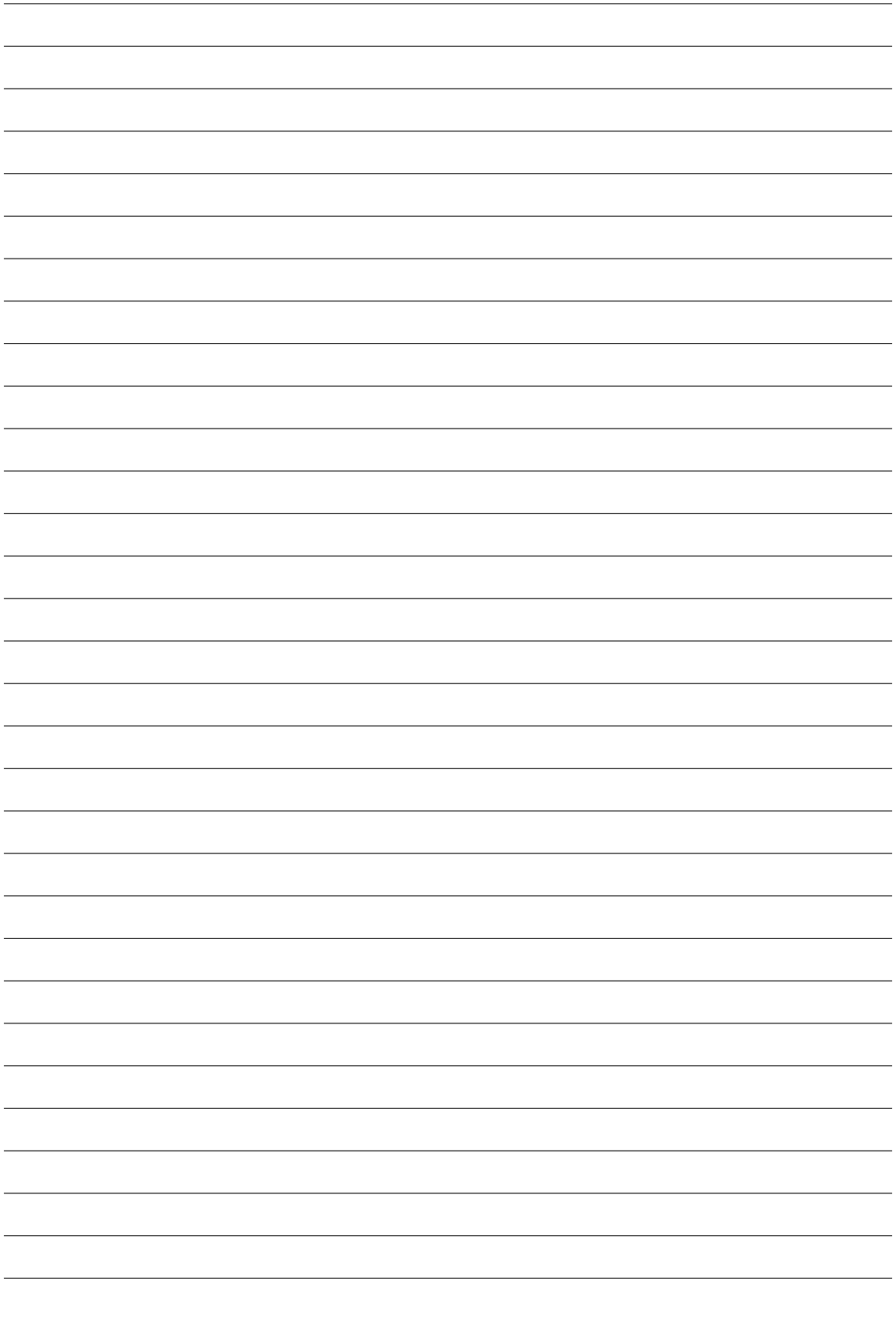
Nom : _____ Prénom : _____

Signature : _____ CIP : _____

Question	Barème
1	/ 20
2	/ 70
3	/ 10
Total :	/100

2. Problème des fumeurs de cigarettes

Soit trois personnes qui fument des cigarettes et un serveur. Chaque fumeur fabrique continuellement des cigarettes et les fume. Pour fabriquer une cigarette et la fumer, un fumeur a besoin de trois ingrédients : le tabac, le papier et une allumette. Chaque fumeur possède seulement un des trois ingrédients en quantité infinie. Ainsi, le premier fumeur possède du tabac, le deuxième possède du papier et le dernier possède des allumettes. Le serveur choisit au hasard deux ingrédients et les met sur la table. Le fumeur qui possède le troisième prend les deux ingrédients sur la table, fabrique une cigarette et la fume. Le serveur attend que le fumeur ait fini et le cycle recommence. Implantez une solution à ce problème basée sur les sémaphores et une autre basée sur les moniteurs.



3. Producteurs/consommateurs (20 points)

Soit la solution suivante au problème des producteurs/consommateurs qui utilise les routines `sleep()` et `wakeup(nom de processus)`, pour effectuer sa synchronisation.

```
#define N = 100
int compte = 0;

void producteur()
{
    int element;

    while (TRUE)
    {
        produire un élément
        if (compte == N) sleep();
        ajouter l'élément dans le tampon
        compte = compte + 1;
        if (compte == 1) wakeup(consommateur);
    }
}

void consommateur()
{
    int element;

    while (TRUE)
    {
        if (compte == 0) sleep();
        enlever un élément du tampon
        compte = compte - 1;
        if (compte == N-1) wakeup(producteur);
        consomme l'élément
    }
}
```

En supposant que :

- les accès à la variable `compte` sont atomiques (addition, soustraction et comparaison)
- les fonctions `sleep` et `wakeup` fonctionnent d'une façon similaire aux fonctions `wait` et `signal` d'une variable condition

Est-ce que cette solution fonctionne ?

Si oui, montrez-le. Si elle ne fonctionne pas, identifiez toutes les sources de problèmes.

