

**Numéro d'anonymat :**

**Examen de IPRM**

**Master Informatique**

**Partie de Cours de S. Chaumette**

**Année 2013-2014**

**Le 8 janvier 2014**

Documents autorisés : aucun

Durée : 50 minutes

Ce sujet comporte 4 pages

**Les réponses seront données sur ce document et uniquement dans les zones prévues à cet effet.**

**Exercice 1.** Cet exercice se présente sous la forme d'un QCM. Pour chaque question, on peut cocher plusieurs cases. Une case cochée à tort comptera des points en moins.

a. On appelle migration forte :

- ☐ un déplacement acquitté par les machines concernées
- ☐ une migration avec reprise sur l'instruction suivant la migration
- ☐ une migration garantie
- ☐ une migration avec reprise sur l'instruction précédant la migration

b. On appelle migration faible :

- ☐ une migration non garantie
- ☐ une migration avec reprise sur l'instruction précédant la migration
- ☐ un déplacement utilisant des GPUs
- ☐ une migration avec reprise sur un début de fonction/méthode
- ☐ une migration avec reprise sur l'instruction suivant la migration
- ☐ une migration avec retour pré-programmé

c. On ne peut pas réaliser d'agents mobiles sans :

- ☐ des *threads*
- ☐ des processeurs
- ☐ le format *xdr*
- ☐ *run-time* Java
- ☐ un mécanisme de sérialisation
- ☐ un mécanisme de type réflexion
- ☐ une connexion réseau

**Numéro d'anonymat :**

**Exercice 2. Numérisation du monde réel.**

On considère un ensemble de 1 milliard de capteurs (ou nœuds) pouvant communiquer dans un rayon maximum de 5 mètres et n'ayant pas d'autre moyen de communication avec l'extérieur.

Question 1. Pourquoi dans un scénario réel ne peut-on pas faire de programme permettant à un ou à tous les nœuds de compter l'ensemble des nœuds du système ?

Question 2. Que peut-on proposer à la place ?

Question 3. Donner un programme (algorithme) qui sera exécuté par chaque capteur permettant de réaliser l'opération que vous avez proposée en réponse à la question 2.

**Numéro d'anonymat :**

**Dans la suite, on fait l'hypothèse que chaque capteur dispose d'un élément sécurisé qui permet de stocker des données et d'exécuter du code qui demeurent inviolables.**

Question 4. Votre programme garantit-il qu'il qu'aucun nœud ne sera compté deux fois ? Si oui, justifier, sinon proposer une solution (algorithme + schéma).

**Numéro d'anonymat :**

### **Exercice 3. Plug-ins**

Expliquer comment réaliser un plug-in en langage C.