	<b>ING2 : EXAMEN DE RATTRAPAGE PROGRAMMATION SYSTEME, RESEAU ET REPARTIE</b>  <b>EXAMEN PAPIER (AUCUN DOCUMENT N'EST AUTORISE)</b>  <b>DUREE 2 HEURES</b>
Rédigé par : équipe pédagogique du cours de programmation système, réseau et répartition.	Ref : <i>ING2-PROG SYS RES PARLL</i>
A l'intention de : Etudiants d'ING2 GSI	Créé le : 16/05/2017

### Exercice 1 (3Pts)


Un élève a donné sa solution à un exercice dont le sujet est le suivant : Ecrire un programme qui remplit le tableau T par le master, puis calcule la somme des éléments du tableau.

```
#include <omp.h>
#include <stdio.h>
#include <stdlib.h>
#define taille 100

int main (int argc, char *argv[]) {
    int i, tid;
    float total;
    float tab[taille];

    /**/ Spawn parallel region /**/
    #pragma omp parallel
    {
        /* Obtain thread number */
        tid = omp_get_thread_num();
        /* Only master thread does this */
        #pragma omp master
        {
            for (i=0; i<taille; i++)
                tab[i]= 0.1*i;
        }
        total = 0.0;
        #pragma omp for schedule(dynamic,10)
        for (i=0; i<taille; i++) {
            total = total + tab[i];
            printf ("Thread %d is done! Total= %e\n", tid, total);
        }
    } /**/ End of parallel region /**/
    printf (" Total global= %e\n", total);
} // end main
```

**Question** : Ce code donne-t-il le résultat voulu ? Sinon comment le corriger ?

	<p align="center"><b>ING2 : EXAMEN DE RATRAPAGE PROGRAMMATION SYSTEME, RESEAU ET REPARTIE</b></p> <p align="center"><b>EXAMEN PAPIER (AUCUN DOCUMENT N'EST AUTORISE)</b></p> <p align="center"><b>DUREE 2 HEURES</b></p>	
Rédigé par : équipe pédagogique du cours de programmation système, réseau et répartition.	Ref : <i>ING2-PROG SYS RES PARLL</i>	
A l'intention de : Etudiants d'ING2 GSI	Créé le : 16/05/2017	

### Exercice 2 (3pts)

Ecrire un programme qui calcule le produit des carrés d'un vecteur V de longueur N connue. L'initialisation du vecteur V se fait par le thread principal.

$$\text{prodCarre} = \prod (x_i * x_i)$$

- a- sur une communication point à point classique
- b- sur une communication point à point sur une structure d'anneau

### Exercice 3 (3pts)

On considère l'architecture logicielle suivante : une application avec un serveur et 2 clients. Le serveur attend des questions de la part des clients. Un client demande au serveur de lui envoyer  $n$  nombres aléatoires compris entre 1 et  $n$ . La valeur ' $n$ ' est générée aléatoirement par un client, et comprise entre 1 et NMAX

1. Faire l'analyse de ce système et donner une architecture en utilisation des tubes ordinaires.
2. Donner le code de ce programme

### Exercice 4 : communication collective (3 pts)

Soit un programme `mpi_allgather.c` qui implémente la primitive `Allgather`. Dans ce programme, chaque processus déclare un tableau T de taille M fixe et un vecteur V de taille  $L=M/N$ , N est le nombre de processus. Chaque processus initialise son vecteur V par la formule suivant

$$V[i] = \text{rang} * L + i$$

Question

1. Ecrire ce programme pour  $M = 128$
2. Pour le nombre de processus N égal à 4, expliquer la sortie de chaque processus de votre programme et en afficher juste la sortie d'un processus quelconque avec son rang et le contenu de son tableau T
3. Par quelles autres fonctions de communication collective peut-on remplacer `Allgather`

### Exercice 5 : (8 pts)

Le « jeu de la vie » est un automate cellulaire bidimensionnel où chaque cellule peut prendre deux valeurs (« 0 » ou « 1 », mais on parle plutôt de « vivante » ou « morte ») et où son état futur est déterminé par son état actuel et par le nombre de cellules vivantes parmi les huit qui l'entourent.

Rédigé par : équipe pédagogique du cours de programmation système, réseau et répartition.

Ref : *ING2-PROG SYS RES PARLL*

A l'intention de : Etudiants d'ING2 GSI

Créé le : 16/05/2017

Chaque cellule compte le nombre de 1 dans son voisinage. Si le nombre total de 1 est pair, la cellule prend l'état 0 sinon, elle prend l'état 1.

On suppose que nous avons la grille initiale ci-dessous, avec les conditions aux limites suivantes constantes. On arrête la simulation lorsque la grille est constante d'une itération à la suivante

1	1	1	1	1	1
1	0	0	0	1	0
1	0	0	1	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	0	0	0	0	0

### Questions

1. Ecrivez un programme en C ou en C++ qui résout simule ce jeu de la vie. (2pts)
2. Parallélisez le programme par des directives OpenMP (3pts)
3. Donnez un algorithme parallèle et le code de cette parallélisation pour une résolution en utilisant MPI. (3pts)