TP 1: La programmation parallèle avec OpenMPI

- **Installation Ubunbtu - Package**

$ sudo apt-get install openmpi

openmpi-common

libopenmpi-dev

- **Directives de compilation et d'exécution :**

mpicc test.c -o test
mpirun -np nb.processus ./test

Exercice 1 : Premier programme (Hello World)
Après avoir installé **OpenMPI** compilez le programme hello.c ci-dessous

```
#include <stdio.h>
#include <mpi.h>
int main(int argc, char *argv[])
{
  int pid, nprocs;
  MPI_Init(&argc, &argv);
  MPI_Comm_rank(MPI_COMM_WORLD, &pid);
  MPI_Comm_size(MPI_COMM_WORLD, &nprocs);
  printf("Bonjour je suis le processus %d sur %d processus \n", pid, nprocs);
  MPI_Finalize();
  return 0;
}
```

Q1. - Exécuter le code ci-dessus avec la commande :

$ mpirun -np 4 ./hello

- Quel est le sens de l'argument -np 4 ?
- Faire varier le nombre de processus. Que peut-on conclure ?

Q2. Modifiez le programme précédent pour que seuls les processus avec un identifiant pair affichent le message 'Bonjour'.

Exercice 2: Échanges entre processeurs

Q1. Ecrire un programme qui permet de faire un échange simple d'un entier entre deux processus (communication point à point).

Q2. Modifier le code de l'exercice précédent pour que :

a. Le processus 0 diffuse un message à tous les autres.
b. Le processus 0 initialise un tableau de n entiers et après le diffuse à tous les processus dans le groupe.
c. Le processus 0 envoi de manière sélective un tableau d'entiers répartie sur n processus afin de calculer la somme des éléments de ce tableau et de retourner le résultat.
d. Le processus 0 envoie chaque ligne d'une matrice de taille n×n respectivement à chaque processus.