

## Final SR03 Printemps 2013

**NOM**

**Prénom**

### PARTIE II (MPI) : 4 points

Soit le code quizz.c suivant :

```
#include <stdio.h>
#include <mpi.h>

int main(int argc, char *argv[])
{
    int rank, size, int_error_handling;
    int val=1, sigma = 0;
    // start MPI
    MPI_Init(&argc, &argv);
    int_error_handling = MPI_Comm_size(MPI_COMM_WORLD, &size);
    int_error_handling = MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    val += rank;
    MPI_Reduce(&val, &sigma, 1, MPI_INT, MPI_SUM, 0, MPI_COMM_WORLD);
    if (rank==0)
        printf("\nresultat %d \n\n", sigma);
    int_error_handling = MPI_Finalize();
    return 0;
}
```

Nous rappelons le synopsis suivant:

int MPI\_Reduce(void \*sendbuf, void \*recvbuf, int count, MPI\_Datatype datatype, MPI\_Op op, int root, MPI\_Comm comm)

Input Parameters :

**sendbuf:** address of send buffer (choice)

**count:** number of elements in send buffer (integer)

**datatype:** data type of elements of send buffer (handle)

**op:** reduce operation (handle)

**root :** rank of root process (integer)

**comm :** communicator (handle)

Q1 : A quoi correspond l'instruction MPI\_Reduce ci-dessus ? Détaillez les arguments et expliquez le fonctionnement de cette primitive dans ce code.

Q2 : Après compilation, si on lance l'exécution avec :  
    `mpiexec -n 6 quizz`  
    le processus de rang 0 affiche : resultat 21  
Est-ce cohérent ? Expliquer.

Q3 : Soit  $k$  le nombre de processus parallèles lancés.  
Donner la formule mathématique qui correspond à ce calcul.

Q4 : Si on remplace `MPI_SUM` par `MPI_PROD`, que calculera le programme ?