

Exercices INF8541: série #2

1. Code Threaded-C de somme (version récursive)

Écrivez la version Threaded-C de la fonction `somme` de l'exercice #1 de la première série d'exercices. Il s'agissait de la fonction qui calculait, de façon récursive et parallèle, la somme des nombres compris entre `i` et `j`.

2. Version Threaded-C de somme (version itérative)

Soit la version suivante de la fonction `somme`:

```
THREADED somme( int i, int j, int *GLOBAL result, SPTR done )
{
    int r, k;

    /* On calcule la somme de l'intervalle de nombres. */
    r = 0;
    for ( k = i; k <= j; k++ )
        r += k;

    /* On retourne le resultat. */
    PUT_SYNC( r, result, done );
    TERMINATE;
}
```

Écrivez la version Threaded-C de la fonction `somme_1_a_n` de l'exercice #2 de la première série d'exercices. Il s'agissait de la fonction qui calculait la somme des nombres compris entre 1 et `n` en créant `P` processus itératifs (où `P` représentait le nombre de processeurs, dans ce cas, on a donc `P = NUM_NODES`) où chaque processus calculait la somme d'un sous-intervalle.

Note: Vous pouvez supposer que les fonction `inf` et `sup` sont disponibles pour calculer les bornes inférieures et supérieures des divers intervalles.