

Exercices INF8541: série #1

1. Parallélisme récursif

En utilisant la notation d'Andrews, écrivez une procédure récursive `somme` qui permette de calculer la somme des entiers compris entre `i` et `j`. Cette procédure doit utiliser une approche de *parallélisme récursif*. Assurez-vous d'utiliser une stratégie diviser-pour-régner *bien balancée*, c'est-à-dire, où les sous-problèmes sont de tailles équivalentes.

Pour calculer la somme des `n` premiers nombres, on pourra alors utiliser la procédure suivante, qui appellera votre procédure avec les arguments appropriés:

```
int somme_1_a_n( int n )
{
    return( somme(1, n) );
}
```

2. Parallélisme itératif

Écrivez une procédure `somme_1_a_n` (et une procédure auxiliaire appropriée), mais en utilisant cette fois une approche de *parallélisme itératif*. De plus, seuls `P` processus auxiliaires peuvent être créés (seulement `P` processeurs sont disponibles). La procédure `somme_1_a_n` doit donc créer ces `P` processus itératifs, chacun de ces processus calculant alors la somme d'un sous-intervalle des nombres compris entre 1 et `n`.

Pour simplifier, vous pouvez utiliser les fonctions auxiliaires suivantes pour déterminer les bornes du sous-intervalle devant être traité par le `i` ème processus auxiliaire:

```
int inf( int i, int n ){ return( i * (n / P) + 1 ); }
# Borne inferieure du i-ieme intervalle.

int sup( int i, int n ){ return( (i+1) * (n / P) ); }
# Borne superieure du i-ieme intervalle.
```