CPSC 418: Parallel Computing
Mini Assignment #2
Due 10:00am on January 13, 2017 in class.

Winter 2016–2017 Term 2

Reduce & Scan

Name: _____ Student Number: _____

Complete your short answers on this sheet. Submit a hardcopy at the beginning of class.

1. **Reduce vs Scan (5 points)**. Assume that we have some data (call it the *input*) distributed over a collection of worker processes (call them the *processes*) and we wish to compute some function of this data to produce some other data (call it the *output*). Our local parallel computing guru has assured us that this function can be implemented with either a single reduce or a single scan operation. Briefly describe what properties of the *input*, *processes* and/or *output* would allow us to determine whether we should use a reduce or a scan.

2. **Generalized reduce (10 points)**. As described in *Lin & Snyder* a generalized reduce is described by four functions: `init()`, `accum()`, `combine()` and `reduceGen()`. Assume that we have 1600 data elements to process and that we set up as balanced a tree as possible.

   (a) If we can give 100 data elements to each leaf process, what is the height of the tree (which is also the maximum number of messages that need to be passed to get between a leaf and the root)? What about if we can give only 60 data elements to each leaf process?

   Height for 100 / leaf: _____ 60 / leaf: _____.

   (b) In the 100 / leaf case, how many times are each of these functions called (over all processes)?

   `init()`: _____ `accum()`: _____ `combine()`: _____ `reduceGen()`: _____.

   (c) In the 60 / leaf case, how many times are each of these functions called (over all processes)?

   `init()`: _____ `accum()`: _____ `combine()`: _____ `reduceGen()`: _____.

3. **Generalized scan (10 points)**. As described in *Lin & Snyder* a generalized scan is described by four functions: `init()`, `accum()`, `combine()` and `scanGen()`. Assume that we have 1600 data elements to process and that we set up as balanced a tree as possible.

   (a) How many times are each of these functions called (over all processes) if we can give 100 data elements to each leaf process?

   `init()`: _____ `accum()`: _____ `combine()`: _____ `scanGen()`: _____.

   (b) How many times are each of these functions called (over all processes) if we can give just 60 data elements to each leaf process?

   `init()`: _____ `accum()`: _____ `combine()`: _____ `scanGen()`: _____.