1. List the members of your group below:

2. Let $\oplus$ denote a associative binary operator and let z denote its identity element (so that $z \oplus x = x$ for all x). Let $A = a_1, \ldots, a_n$ be an array of elements from the domain of $\oplus$. We define the *reduce* operation as $R(\oplus, I, A) = a_1 \oplus a_2 \oplus \cdots \oplus a_n$ for $n \geq 0$ (so that $R(\oplus, I, A) = 0$ if $n = 0$).

   Compute $R(\oplus, I, A_1)$ where $A_1 = (3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5)$ and and $\oplus$ is (1) the usual integer addition and (2) string concatenation. What is z for these two cases?

3. The result of the *all prefix sums*, or *scan*, operator applied to a sequence of numbers $a_1, \ldots, a_n$ is the sequence $b_1, \ldots, b_n$, where $b_i = \sum_{j=1}^{i} a_i$ for $1 \leq i \leq n$. (The operator may be generalized beyond addition to any associative operator $\oplus$, as in Question 2.) The *prescan* operator is similar, but returns the sequence $0, b_1, b_2, b_{n-1}$ instead.

   Compute the scan and prescan of the array A_1 of Question 2.

4. Provide a naive $O(n)$ sequential algorithm for *reduce*.

5. Provide a parallel algorithm for reduce that runs in $O(\log n)$ time. How many processors does your algorithm require, as a function of n? Justify the correctness and running time of your algorithm briefly.

6. Modify your algorithm of Question 5 to use a parameterized number of processors, p, yielding a running time of $O(n/p + \log p)$. Justify the correctness and running times of your algorithm.

7. Repeat Questions 4–6 for the *scan* operation.