1. In the global sum problem that we discussed in class (and in Section 1.3 of the textbook), if we assume that there is a variable called \( my\_rank \) (local to each core) that gives each core a unique rank from 0 to \( p-1 \) (for \( p \) cores), devise an expression to calculate \( my\_first\_i \) and \( my\_last\_i \) assuming \( n \) is divisible by \( p \) and \( n > p \).

2. Repeat the above problem if \( n \) is not divisible by \( p \).

3. We have seen two ways of calculating the final sum in the global sum example. In one of them, the master core receives the partial sums from the other cores and calculates the final sum. The other method is the tree-method. Assume that the master core is core 0.
   a. Derive a formula for the number of receives and additions that core 0 does in the first (non-tree) method.
   b. Repeat for the tree-method.
   c. Make a table showing the number of receives and additions done by core 0 for each method when the number of cores is 2, 4, 8, ..., 1024.
   d. Which operation do you think is more expensive: receive or addition? and why?