1 Question 1: Implementing a distributed reduction

Done as instructed and submitted.
Two advantages of a tree-based reduction scheme as implemented in e) are:

- Numerical efficiency: Tree-based reduction has $\mathcal{O}(\log n)$ additions instead of $\mathcal{O}(n)$ for naive implementation
- Bandwidth efficiency: Likewise, less data has to be exchanged on the bus.

2 Question 2: MPI Bug Hunt

2.1 a)

There is no MPI code in this snippet. In case the the calculation part was meant to be the MPI component, all setup and finalize statements would be missing.

2.2 b)

This will cause a deadlock as both rank 0 and rank 1 will first attempt to send their message with the respective partner not ready to receive. This can be fixed by moving the `MPI_Recv` into the `if... else` construct and making sure one rank first receives and then sends while the other rank first sends and then receives.

2.3 c)

If there's only one rank, the program will print `[0]0` to `std::cout`. With more than one rank, the program will also print the above but then hang endlessly, as `MPI_Recv` isn't the proper way to receive broadcast data: If a broadcast is desired, all ranks should call `MPI_Bcast()` and then all ranks end up with their buffer equal to the buffer of the `root` argument of the function call.