1 MPI Hands-On; Sending and Receiving Messages II

1. **Sending in a ring**. A code27.c that takes data from process zero and sends it to all of the other processes by sending it in a ring.

    - That is, process **i** should receive the data and send it to process **i+1**, until the last process is reached.
    - Assume that the data consists of a single integer. Process zero reads the data from the user.

2. The code3.c consists of one receiver process and N-1 sender processes.

    - The sender processes send a message consisting of their process identifier (id) and the total number of processes (ntasks) to the receiver.
    - The receiver process prints out the values it receives in the messeges from the senders.

3. Analyse the example code2 for sending/receiving.

4. **Exercise**: Write a complete program to make a parallel multiplication of two arrays (a and b) with the size of one million each.

$$c = \sum_{i=1}^{1000000} a[i] * b[i]$$

    - Fill the arrays with random numbers.
    - Master node should distribute the date to the nodes.
    - Your code should work both for even and odd numbers of compute nodes.
    - Master node should also be involved to the action.
    - There should be element-wise multiplication.
    - Each node also should have a local sum and send to master node their values to added by master node up.
    - Master node should print out the final summation value (c) and also report the local values (c at nodes)with corresponding nodes.