IN3200/IN4200 Exercise Set 10

Exercise 1

Write a new Hello World program, where all the processes first generate a text message using `sprintf` and then send it to Process 0. (Hint: you may use `strlen(message)+1` to find out the length of the message). Afterwards, Process 0 is responsible for writing out all the messages on the standard output.

Exercise 2

Write three simple parallel programs for adding up a number of random numbers. Each process should first generate and sum up locally an assigned number of random numbers. To find the total sum among all the processes, there are three options:

- Option 1: let one process be the master and let each process use `MPI_Send` to send its local sum to the master.
- Option 2: let one process be the master such that it collects from all the other processes by the `MPI_Gather` command.
- Option 3: let one process be the master and make use of the `MPI_Reduce` command.

Exercise 3

The computation to be considered is the dense matrix-vector multiplication

$$\mathbf{y} = \mathbf{A}\mathbf{x}$$

where $\mathbf{A}$ is an $N \times N$ matrix, $\mathbf{x}$ is the input vector and $\mathbf{y}$ is the result vector.

Please divide the work following a row-wise decomposition of $\mathbf{A}$, and write a corresponding MPI program. To begin with, process with rank 0 is supposed to have the entire content of $\mathbf{A}$ and $\mathbf{x}$. Each of the the other processes has to communicate with rank 0 to acquire $\mathbf{x}$ and its designated piece of $\mathbf{A}$. When all the processes finish their local computation, MPI communication is needed to let rank 0 gather the entire result vector $\mathbf{y}$.

Note: N is not necessarily divisible by the number of processes.

Exercise 4

Repeat the above exercise, but using instead a column-wise decomposition of matrix $\mathbf{A}$ (which is distributed to the other MPI processes).