# PG2105 Heterogeneous Computing for AI

## Hands-on exercises for week 04 (2022-09-23)

## Raghava Mukkamala (rrm.digi@cbs.dk)

**Exercise 1:**

```
Use the in class 'ex_01_range_counter.py' and vary the size of input data.

Record the time it takes to run the sequential and parallel versions and
plot them in a graph. Plot them for the following sizes:
[50000,500000,900000,3000000, 3000000,9000000]

Example: np.random.randint(0, 10, size=[50000, 10]) for the first value.
```

**Exercise 2:**

```
Create a function that returns the cube of the sum of all numbers in the
row. Use the in class 'ex_01_range_counter.py' and vary the size of input
data. Record the time it takes to run the sequential and parallel versions
and plot them in a graph.
```

**Exercise 3:**

```
Use the code from the python file 'ex_03_pipes.py' where a consumer and a
producer run on separate processes and communicate using a pipe. You can
see that producer is sending data (list of integers) to the consumer in
the sample code.

Rewrite the application such that there is two-way communication between
the producer and consumer. More specifically, the consumer will calculate
the square of numbers sent by the producer and send the list back to the
producer. For example, if the consumer receives [1,2,3..], then the
consumer will send [1,4,9,..] back to the producer.
```

**Exercise 4:**

```
Use the code from the python file 'ex_04_producer_consumer_queue.py' where
a consumer and a producer run on separate processes and communicate using
a queue. You can see that producer is sending data (random values) to the
consumer in the sample code.
```

```
Rewrite the application such that there is two-way communication between
the producer and consumer. More specifically, the consumer will calculate
the square of numbers sent by the producer and send the list back to the
producer. For example, if the producer send a list, e.g. [1,2,3..], the
consumer receives it and then the consumer will send [1,4,9,..] back to
the producer, similar to the exercise in 3.
```