Assignment 5, due April 12th 2019

Assignment:

- Create a sequential C program which
  - Takes an input parameter N
  - Creates a byte array of N elements
  - Fills this array with random numbers 0/1
  - Computes the number of entries that hold "1"
- Parallelize the computation step with OpenMP
- Parallelize the computation step with OpenCL using recursive multi-stage reduction as discussed in the lecture
- Test your implementations by comparing the results for multiple N and random seeds
- Compare the performance of the sequential, OpenMP and OpenCL computation for multiple N such that 1, 2, 3, … stages are necessary until reaching the GPU global memory limit.

Hints:

- Random numbers:
  - Use `rand()` in `stdlib.h` for getting random numbers
  - For getting multiple test cases, vary the random seed (`srand`)
- Performance comparison:
  - Compile the sequential and OpenMP versions with full compiler optimizations
  - For the OpenCL implementation, only measure the computation time on the device, not the data transfer

Goal:

- Code development without a template
- Implementation of a reduction in OpenCL

Solution upload:

- Via e-mail to philipp.gschwandtner@uibk.ac.at – one submission per group only!
  Subject: "`[PS703106] [AS05] GR_## - NAME1, NAME2, NAME3`"
  Solution must be submitted before Friday 09:15!