Task 5.1 Preparation

- Install a debugger such as *gdb* or *lldb*, optionally within your IDE.

Task 5.2 PennaLV

- Write a simulation class to reduce the code in the main.cpp. Remove `N_max`, `N_init` and `N_t` (population properties) from the `species_property_struct`, since these variables aren't really a property of the animals. The new simulation class should take care of counting and managing the population. The configuration (both animal properties and population properties) should still be done in main.cpp. Some signature changes in the animals will be necessary (to provide `N_t`). Use your test framework to ensure that the results don't change.

Task 5.3 Debugging

- Build a debug version of your pennaLV and use a debugger of your choice to investigate it:

	gdb	lldb
launch the debugger and load the binary	`gdb ./path/to/main`	`lldb ./path/to/main`
set a break point in `sheep.cpp`	`break filename:line`	`b filename:line`
run it up to the break point	`run`	`run`
read the privates of bear!	`print bear::xyz`	`print bear::xyz`

- Find out the current lower/upper parameters of the bear RNG distribution (*hint: stored as* `_M_a`, `_M_b`)
- Try to change a constant or static constant class member
- Cheat sheet: http://lldb.llvm.org/lldb-gdb.html

Task 5.4 Challenge (optional)

This week there is no challenge (chocolate shortage :-P).