Exercise 1

Part 1

In this exercise, you will compile your first program for the Xeon Phi accelerator cards. Write a program that computes the *AXPY function of the BLAS API and evaluates this function on two static arrays, in order to achieve maximum theoretical performance.
Therefore, download the file u1.tar.bz2 from CAJ and extract the content to your home directory on the lab server: **raj.inf-i2.uni-jena.de**.

One you have done that, you will find a directory u1 in your home. Create a directory called *build* within the extracted directory and change into that directory. Afterwards, execute the following commands:

Example:
$> `CC=icc CXX=icpc cmake ..`
$> `make`

In contrast to to a regular call to cmake, these calls tell CMake to use the Intel Compiler instead of the default gcc/g++. Have a look at the CMakeLists.txt that you downloaded. You will find an option `TARGET_MIC`. How is this this option used within the script? Use separate calls of cmake to switch between building for the host and building for the accelerator cards.

Example: Configures the Makefile to build a binary that runs on the host
$> `cmake -DTARGET_MIC=OFF ..`

At this time your program does no even compile. You can check this yourself by calling `make`. Have a look at the file axpy.cpp and edit the TODOs within until your program both compiles and runs correctly.
In particular, calculate the FLOPS your program performs, and don't forget to set the `#define` Makros at the beginning of the file in order to do that (Look for the ??? markers).

Try to guess before the first run: How many GLFOP/s can you expect? Consider in the context of this exercise, that your program is running single-threaded only. Review the slides from last week to find your answer. After compiling for the host, run your program as usual. After compiling for the cards, you have to login to one of the first before you can run your program:

Example: Login to card 2 of 4, which has index 1
$> `ssh mic1`

Navigate to your build directory and execute your binary. Does your implementation meet your expectations when running on the cards?