- In week 40 we started with "Introduction to Parallel Machines and Programming Models". We also started with "Shared Memory Programming: Threads and OpenMP" (Chapter 7 in the course book covers that topic, but I will not strictly follow the book).
- In week 39/40 you got a brief introduction to Streaming SIMD extensions (SSE). Here is a simple example code to compute a inner product of two vectors x and y, the code presented in the lecture can be found in blackboard:

```
#include <pmmintrin.h>
 int main()
{
    int k,i;
    k = 100;
    float x[k]; float y[k]; // vectors of length k
    __m128 X, Y;
    // 128-bit values
    __m128 acc = _mm_setzero_ps(); // set to (0, 0, 0, 0)
    float inner_prod, temp[4];
    for(i = 0; i < k - 4; i += 4) {
        X = _mm_load_ps(&x[i]); // load chunk of 4 floats
        Y = _mm_load_ps(y + i); // alternate way, pointer arithmetic
        acc = _mm_add_ps(acc, _mm_mul_ps(X, Y));
    }
    _mm_store_ps(&temp[0], acc); // store acc into an array of floats
    inner_prod = temp[0] + temp[1] + temp[2] + temp[3];
    // add the remaining values
    for(; i < k; i++)
        inner_prod += x[i] * y[i];
}
```

- For working on the first assignment I highly recommend to use `http://godbolt.org/` in order to see how optimization flags change your code, how compilers change your code, and so on.
- In week 41 we will start with "Distributed Memory Machines and Programming" including MPI (Chapters 6 and 7 in the course book). After that we will start with Chapter 3 of the course book.
- Note also that you are responsible if you use too much computation time on the supercomputer(s), and that you will not get additional computation time, in case you run out of computation hours. Please check the NIM web interface at `http://www.nersc.gov/nusers/accounts/nim/`.

- In contrast to the method on Edison (see for example `http://www.nersc.gov/users/computational-systems/edison/running-jobs/`), MPI programs are started with `mpirun` on the IMADA pool. This will also be discussed in week 41. You have to make sure that it is possible to login to all machines you want to use via `ssh` without being asked for your password. This can be accomplished with ssh keys as follows. SSH keys allow authentication between two hosts without the need of a password. SSH key authentication uses two keys a private key and a public key. To generate the keys, from a terminal prompt enter:

  ```
  ssh-keygen -t dsa
  ```

  This will generate keys using a DSA authentication identity of the user. During the process you will be prompted for a password. Do not enter a password. Simply hit Enter when prompted to create the key. By default the public key is saved in the file `~/.ssh/id_dsa.pub`, while `~/.ssh/id_dsa` is the private key. Now append `id_dsa.pub` to `~/.ssh/authorized_keys2`. (On the IMADA pool all machines use the same file system, therefore this is not a remote operation. If you want to login to another remote machine without being asked for the password, you have to append the key to the corresponding file on the remote machine):

  ```
  cat id_dsa.pub >> .ssh/authorized_keys2
  ```

  Finally, double check the permissions on the `authorized_keys2` file, only the authenticated user should have read and write permissions. If the permissions are not correct change them by:

  ```
  chmod 644 .ssh/authorized_keys2
  ```

  You should now be able to SSH to the host without being prompted for a password. Note that you have to login to all machines that you want to use once manually and answer the question

  ```
  Are you sure you want to continue connecting (yes/no)?
  ```

  once.