David Uhlfelder and Anurag Kuppala
CPE 315 Lab 9

1. If you are building a processor and have to do static branch prediction (meaning you have to assume at compile time whether a branch is taken or not), how should you do it?

   Static branch prediction determines whether or not a branch is taken by the branch instruction rather than the actual actions of the code. The first step in this is to assume every branch jump is not taken in order to fetch every instruction. Then, upon evaluation, the pointer will reach a non-sequential address only if a jump is found to be taken.

2. If you are building a 256-byte direct-mapped cache, what should you choose as your block (line) size?

   The optimal byte block size for higher levels of optimization was 32 bytes, but it was lower (8 bytes) for lower levels of optimization. I would pick a 32 byte block size because I would want to run my program at a high optimization level.

3. What conclusions can you draw about the differences between compiling with no optimization and -O2 optimization?

   O2 optimization leads to a much faster runtime overall. This can be attributed to a few factors. Mainly, the number of dynamic instructions is about 5 times more without optimization, forcing the program to go through a lot more instructions, and while no optimization has a higher hit rate, this still doesn't make up the difference.