Team Members:
Kunal Choudhary, 999981863, kunal.choudhary@mail.utoronto.ca;
Imran Ariffin, 1000863176, imran.ariffin@mail.utoronto;

Try1: par_block.h par_block.c
Implementation:
Split the board into nthread blocks, with one thread to process each block.

optimizations used:
pthread with 8 threads
instruction level parallelism for neighbour_count
reduce usage of mod function to only at margins of block

Try2: par_blk_barrier.h, par_blk_barrier.c
implementation:
Same as Try1, except using barrier instead of join.

optimizations used:
pthread with 8 threads
instruction level parallelism for neighbour_count
wait on barrier instead of join

Try3: par_blk_margin.h, par_blk_margin.c
Implementation:
Same as Try1, except margin is left to be processed by an extra designated thread.

optimizations used:
pthread with 8 + 1 threads (+ 1 for margin)
instruction level parallelism for neighbour_count
margin of the whole board calculated separately by a designated thread. This helps to get rid of modulo op for most cells and reduce cache misses at margin.