HW2

Due: Mon., March 25th

1. Write sequential and multi-core parallel programs for Mandelbrot Set. Measure the execution time as $K$ changes from 1 to 4, and compute speedup and efficiency.

2. Do the same as Q1 for estimating PI.

3. Do the same as Q1 for the cellular automata problem. You need to do barrier actions.

4. We know that we can compute $\cos(x)$ using the Taylor series expansion. Answer the following questions on this method:
   - Write a sequential program that computes $\cos(x)$ for $x = 0.0, 0.1, 0.2, ..., 14999.9, 15000.0$ with three digits of precision. In other words, the Taylor series expansion should continue until $|\text{term/sum}| < 0.001$ where term refers to the last term in the Taylor series and sum is the summation of all the terms expanded so far.
   - Convert the sequential program into an SMP parallel program (in pseudocode).
   - Calculate Speedup and Eff as $K$ (number of processors) changes.

5. The binomial coefficient \( \binom{n}{k} \) is the coefficient of the $x^k$ term in the polynomial expansion of $(1+x)^n$. The binomial coefficients for various values of $n$ and $k$ are the entries of Pascal’s Triangle.
   
   (a) Computing each entry is computationally expensive. What is an alternative computationally more efficient way in which the entries of row $i+1$ is computed from those of row $i$?

   (b) Write a sequential program that computes Pascal’s Triangle in which the entries of row $i+1$ is computed from those of row $i$.

   (c) Convert your sequential program into a parallel program, and compute the speedup and efficiency.