```cpp
//////////////////////////////
//// Kenneth -- Parallel Expectimax
////    for details on the problem and expectimax algorithm refer to expectimax.h
//// WARNING -- The problem size that is being worked on is 4^n + 4^n-1 + ... + 4^1 + 1
////            so be aware that even a deceptively small algorithm depth could result in
////            waiting until the end of the universe to complete.
//// compile and run:
//// $> g++ -fopenmp -o xxx final-project.cpp
//// $>./xxx <number of threads> <depth of algorithm>
//////////////////////////////
#include<iostream>
#include<omp.h>
#include<cstdlib>
using namespace std;
#include"world.h"
#include"expectimax.h"
#include"driver.h"


int main(int argc, char** argv){
    // take in thread count and depth as system arguments.
    int thread_count = atoi(argv[1]);
    int depth = atoi(argv[2]);
    BenchmarkSerialAlgorithm(depth);
    BenchmarkParallelAlgorithm(depth,thread_count);
}
```