#include <iostream>

#include <fstream>

#include <cstdlib>

#include <cmath>

#include "spinarray.h"

using namespace std;

//Simulation of a 2D 20x20 array of spins under changing temperature with static external magnetic field

int main(void)

{

ofstream results ("results.txt"); //output file for E,M,C,X

ofstream pattern ("pattern.txt"); //output file for spins

//Initialisations

spinarray spin; //array of spins

double uB = -0.5; //product of Bohr magneton and external magnetic field

double E = -400.0 - uB\*400.0; //energy

double Echg = 0.0; //change in energy

double S = 400.0; //sum of spins

double Schg = 0.0; //change in sum of spins

srand(100); //seed random number generator

//Main code

for (double beta=1.0; beta>=0.0; beta-=0.01) //varying temperature coefficient beta

{

double trackE[150]; //array of evolution of energy

double trackS[150]; //array of evolution of sum of spins

for (int h=1; h>=1 ; h+=1)

{

trackE[(h+150)%150] = E; //record progression of energy

trackS[(h+150)%150] = S; //record progression of sum of spins

double lowavgE = 0.0; //energy average of the 1-50 of the latest 150 states

double midavgE = 0.0; //energy average of the 51-100 of the latest 150 states

double upavgE = 0.0; //energy average of the 101-150 of the latest 150 states

if (((h+150)%150)==0){

for (int n=0; n<50; n+=1) //calculate moving averages of energy

{

lowavgE+=trackE[n]/50.0;

midavgE+=trackE[n+50]/50.0;

upavgE+=trackE[n+100]/50.0;

}

cout << h << '\t' << lowavgE << '\t' << midavgE << '\t' << upavgE << endl; //code to track progress of program

//Determine steady state

if ( fabs(lowavgE - midavgE) < 0.5 ){ //stability condition

if ( fabs(midavgE - upavgE) < 0.5 ){ //stability condition

h = h\*(-1); //establishing condition for stopping loop

double Esum = 0.0; //sum of total energies of latest 150 states

double E2sum = 0.0; //sum of total energy squared of latest 150 states

double Ssum = 0.0; //sum of total spins of latest 150 states

double S2sum = 0.0; //sum of total spin squared of latest 150 states

// Calculate Esum, E2sum, Ssum, S2sum

for (int n=0; n<150; n+=1)

{

Esum += trackE[n];

E2sum += trackE[n]\*trackE[n];

Ssum += trackS[n];

S2sum += trackS[n]\*trackS[n];

}

//Determine state variables of steady state

double Eavg = Esum/150.0; //energy

double E2avg = E2sum/150.0;

double Savg = Ssum/150.0;

double S2avg = S2sum/150.0;

double M = Savg/400.0; //magnetisation

double C = beta\*beta\*(E2avg - Eavg\*Eavg)/400.0; //specific heat capacity

double X = beta\*(S2avg - Savg\*Savg)/400.0; //magnetic susceptibility

//Output state variables

results << beta << '\t' << Eavg << '\t' << M << '\t' << C << '\t' << X << endl;

//Determine distribution of spins

double spinavg = 0.0; //average spin

for (int i=1; i<=20; i+=1)

{

for (int j=1; j<=20; j+=1)

{

spinavg+=spin.get(i,j)/400.0;

}

}

double spinvar = 1.0 - spinavg\*spinavg; //variance of spin

//Output distribution of spins

pattern << beta << '\t' << spinavg << '\t' << spinvar <<endl;

//Reset tracker for next temperature

for (int n=0; n<150; n+=1)

{

trackE[n] = 0.0;

trackS[n] = 0.0;

}

}

}

}

//Select random spin

double x, y; //random doubles between 1 and 20

int i, j; //location of randomly selected spin

x = 19.0\*rand()/RAND\_MAX + 1.0; //setting x

y = 19.0\*rand()/RAND\_MAX + 1.0; //setting y

i = int(x); //assigning random integer

j = int(y); //assigning random integer

//Selecting neighbours of random spin, considering periodic boundary conditions

// l, r: x-coord of left, right neighbouring spins; d, u: y-coord of down, up neibouring spins

int l, r, d, u;

if (i==1){

l = 20;

}else{

l = i-1;

}

if (i==20){

r = 1;

}else{

r = i+1;

}

if (j==1){

d = 20;

}else{

d = j-1;

}

if (j==20){

u = 1;

}else{

u = j+1;

}

//Calculating change variables

double s = spin.get(l,j) + spin.get(r,j) + spin.get(i,d) + spin.get(i,u); //sum of neighbouring spins

Schg = -2.0\*(spin.get(i,j)); //change of sum of spins if selected spin flips

Echg = (spin.get(i,j))\*2.0\*s - uB\*Schg; //change of energy if selected spin flips

double flip = -1.0\*(spin.get(i,j)); //spin of flipped spin

//Determining evolution of system and updating state

double det = 1.0\*rand()/RAND\_MAX; //random double between 0 and 1

double ex = exp(-Echg\*beta); //exponential for determing flip

if (Echg<0.0){

spin.set(i,j,flip);

E+=Echg;

S+=Schg;

}else{

if(det<ex){

spin.set(i,j,flip);

E+=Echg;

S+=Schg;

}

}

}

}

}