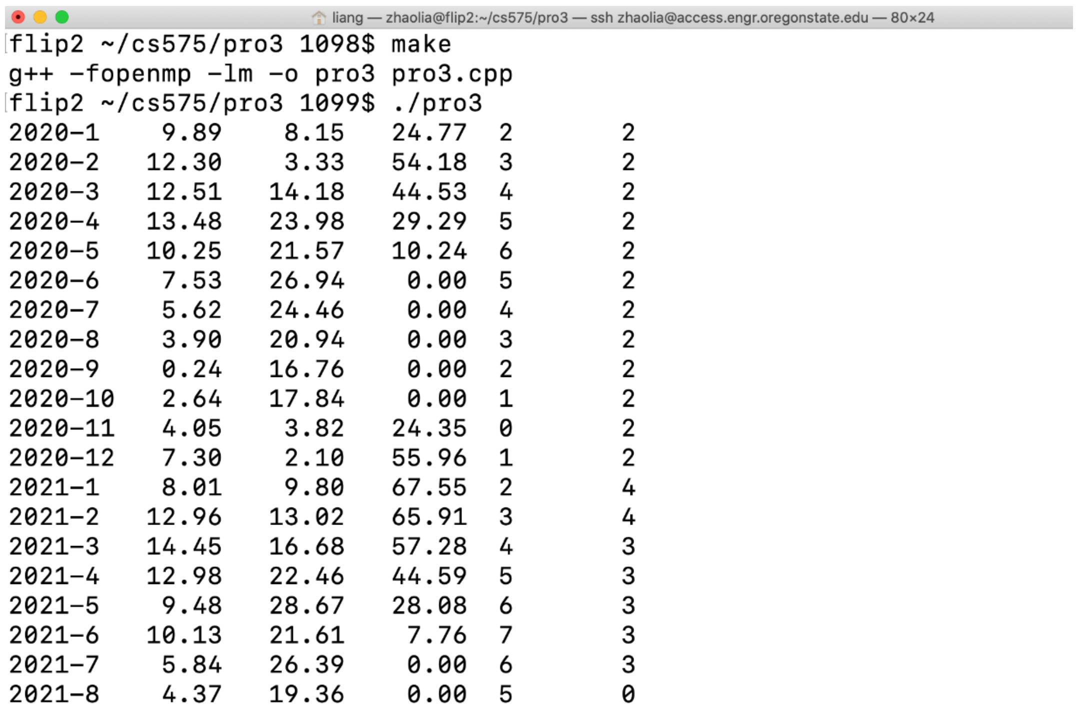
CS 475/575 -- Spring Quarter 2020

Project #3

Functional Decomposition

100 Points

Due: May 4


liang — zhaolia@flip2:~/cs575/pro3 — ssh zhaolia@access.engr.oregonstate.edu — 80×24

```
flip2 ~/cs575/pro3 1098$ make
g++ -fopenmp -lm -o pro3 pro3.cpp
flip2 ~/cs575/pro3 1099$ ./pro3
2020-1     9.89     8.15    24.77  2       2
2020-2    12.30     3.33    54.18  3       2
2020-3    12.51    14.18    44.53  4       2
2020-4    13.48    23.98    29.29  5       2
2020-5    10.25    21.57    10.24  6       2
2020-6     7.53    26.94     0.00  5       2
2020-7     5.62    24.46     0.00  4       2
2020-8     3.90    20.94     0.00  3       2
2020-9     0.24    16.76     0.00  2       2
2020-10    2.64    17.84     0.00  1       2
2020-11    4.05     3.82    24.35  0       2
2020-12    7.30     2.10    55.96  1       2
2021-1     8.01     9.80    67.55  2       4
2021-2    12.96    13.02    65.91  3       4
2021-3    14.45    16.68    57.28  4       3
2021-4    12.98    22.46    44.59  5       3
2021-5     9.48    28.67    28.08  6       3
2021-6    10.13    21.61     7.76  7       3
2021-7     5.84    26.39     0.00  6       3
2021-8     4.37    19.36     0.00  5       0
```

Liang Zhao

933-667-879

zhaolia@oregonstate.edu

Hi there, this project is exciting to simulate the growth of grain by using parallelism. Grain grows affected by the temperature, amount of precipitation, and the number of deer around to eat it, and GrainMoods, which is a global variable that I set myself. When the grain is in a good mood, it grows fast, and when it is in a bad mood, it grows slowly. My program runs on the OSU Linux server: access.engr.oregonstate.edu. I used a bash script to collect all the data. Let us first look at the data in the table, after which I will explain the specific meaning.

Time(Year-Month)	Precipitation	Temperature(°C)	Grain's Height(cm)	Deer	GrainMoods
2020-1	9.89	8.15	24.77	2	2
2020-2	12.3	3.33	54.18	3	2
2020-3	12.51	14.18	44.53	4	2
2020-4	13.48	23.98	29.29	5	2
2020-5	10.25	21.57	10.24	6	2
2020-6	7.53	26.94	0	5	2
2020-7	5.62	24.46	0	4	2
2020-8	3.9	20.94	0	3	2
2020-9	0.24	16.76	0	2	2
2020-10	2.64	17.84	0	1	2
2020-11	4.05	3.82	24.35	0	2
2020-12	7.3	2.1	55.96	1	2
2021-1	8.01	9.8	67.55	2	4
2021-2	12.96	13.02	65.91	3	4
2021-3	14.45	16.68	57.28	4	3
2021-4	12.98	22.46	44.59	5	3
2021-5	9.48	28.67	28.08	6	3
2021-6	10.13	21.61	7.76	7	3
2021-7	5.84	26.39	0	6	3
2021-8	4.37	19.36	0	5	0
2021-9	1.72	19.2	0	4	0
2021-10	1.72	14	0	3	4
2021-11	3.24	13.07	0	2	4
2021-12	6.44	2.24	25.52	1	3
2022-1	10.42	7.58	53.76	2	4
2022-2	12.3	12.14	54.34	3	4
2022-3	12.71	13.36	48.33	4	4

2022-4	12.58	20.42	35.64	5	3
2022-5	10.78	28.38	19.13	6	3
2022-6	10.47	23.24	0	7	3
2022-7	4.46	26.07	0	6	3
2022-8	2.08	18.69	0	5	0
2022-9	0.91	19.49	0	4	0
2022-10	0.54	17.91	0	3	0
2022-11	2.37	7.71	4.65	2	4
2022-12	3.49	4.97	25.93	1	0
2023-1	9.3	6.78	56	2	4
2023-2	11.54	9.26	69.67	3	4
2023-3	13.21	12.33	65.68	4	4
2023-4	13.57	18	53.07	5	3
2023-5	11.67	21.01	36.57	6	3
2023-6	8.32	31.07	16.25	7	3
2023-7	5.57	30.41	0	6	3
2023-8	2.71	27.43	0	5	0
2023-9	0.45	15.03	0	4	4
2023-10	3.16	15.95	0	3	4
2023-11	3.53	6.68	13.86	2	4
2023-12	6.31	1.44	35.24	3	3
2024-1	8.13	3.95	65.29	4	2
2024-2	12.8	7.78	76.27	5	2
2024-3	11.75	13.26	60.4	6	2
2024-4	11.73	18.89	37.59	7	2
2024-5	12.26	26.56	10.92	8	2
2024-6	9.57	28.72	0	7	2
2024-7	6.48	21.91	0	6	2
2024-8	3.7	25	0	5	2
2024-9	0.77	18.01	0	4	2
2024-10	0.99	12.98	0	3	2
2024-11	1.24	10.73	0	2	2
2024-12	3.74	9.98	2.57	1	2
2025-1	7.47	5.59	35.29	2	4
2025-2	10.77	8.07	56.59	3	4
2025-3	11.82	15.02	48.75	4	4
2025-4	13.06	21.32	36.05	5	3
2025-5	12.92	25.67	19.54	6	3
2025-6	9.97	30.83	0	7	3
2025-7	7.42	21.87	0	6	3

2025-8	1.99	19.33	0	5	0
2025-9	1.87	18.65	0	4	0
2025-10	0.58	12.17	0	3	4
2025-11	4.19	10.08	1.47	2	4
2025-12	5.39	0.36	15.52	1	3

Since the globally recognized scientific units of measurement are degrees Celsius and metric units, so I used "° C" and "cm". Sometimes due to many deer, all the grains are eaten up. I also try to increase the amount of grain that grows every day and reduce the amount of deer that I eat every day. Let's take a look at my own-choice quantity.

I use GrainMoods to express the mood of grains. I know that plants have no emotions. I am just for fun, lol. The default mood value of grains is 1. When encountering different years or months or for different temperatures, the mood of grain will increase. When the temperature is too low, the value of encouraging feelings will decrease. Just like people's moods will change. So when the mood value of the grain is greater than 6, it will increase by 2 inches per month, and when it is greater than 3, it will increase by 1 inch per month. If it is less than 3, it is called a bad mood, and it will not increase much, just according to the temperature, and precipitation grows.

My code is shown below:

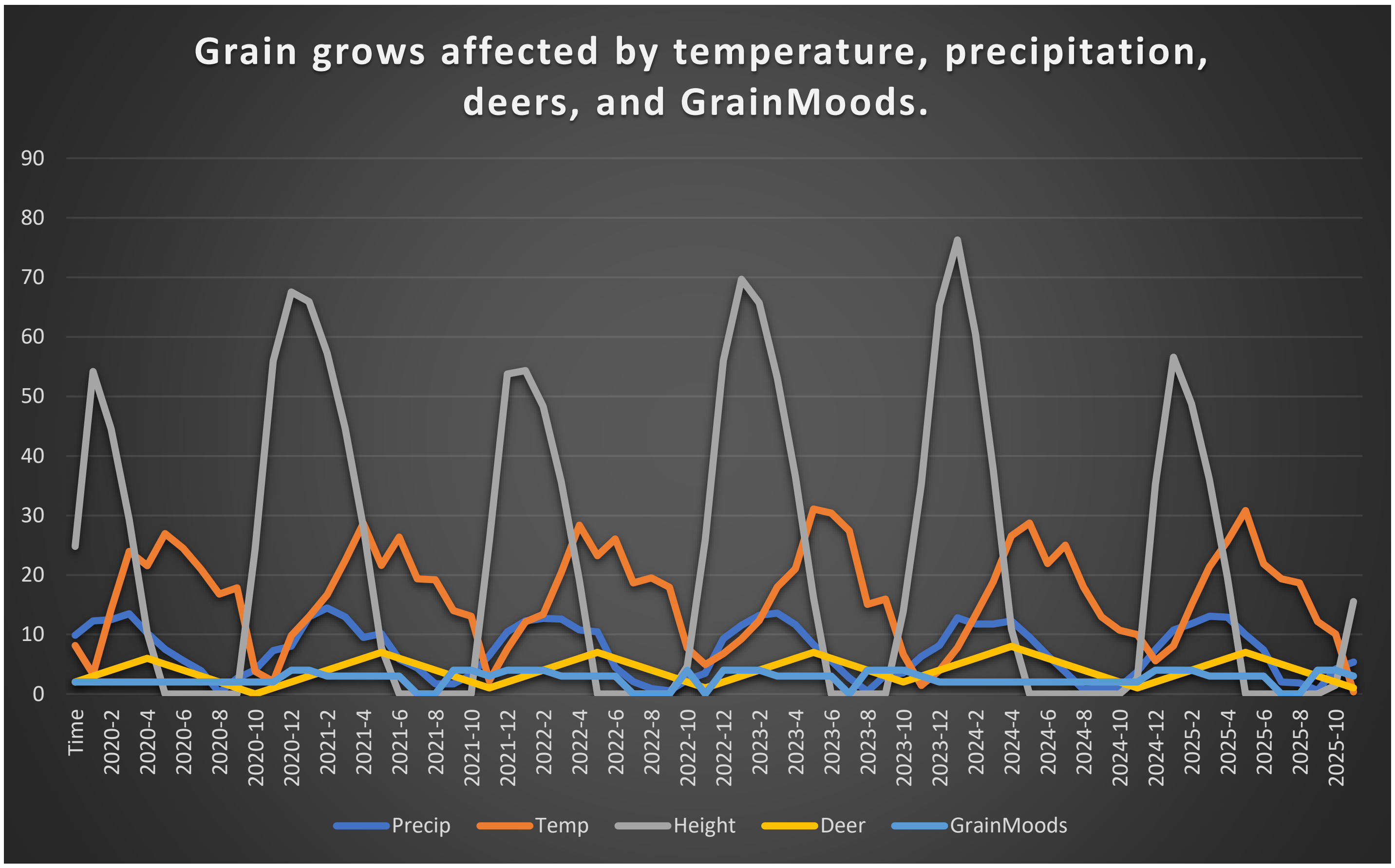
```
    int mood = GrainMoods;

    mood = 1;
    //Calculate mood index

    if(NowYear % 400 == 0 || (NowYear % 4 == 0 && NowYear % 100 != 0))
        mood += 1;
    else if(NowTemp >= 42.0 && NowTemp <=62.0 )
        mood +=3;
    else if(NowPrecip > 5)
        mood += 2;
    else if(NowMonth == 5 || NowMonth == 6)
```

```
        mood += 2;
    else if (NowMonth == 11 &&NowTemp <= 50 )
        mood -=1 ;
    else
        mood -= 2;


    if (mood <= 0)
         mood = 0;
```

Grain’s mood affects growth:

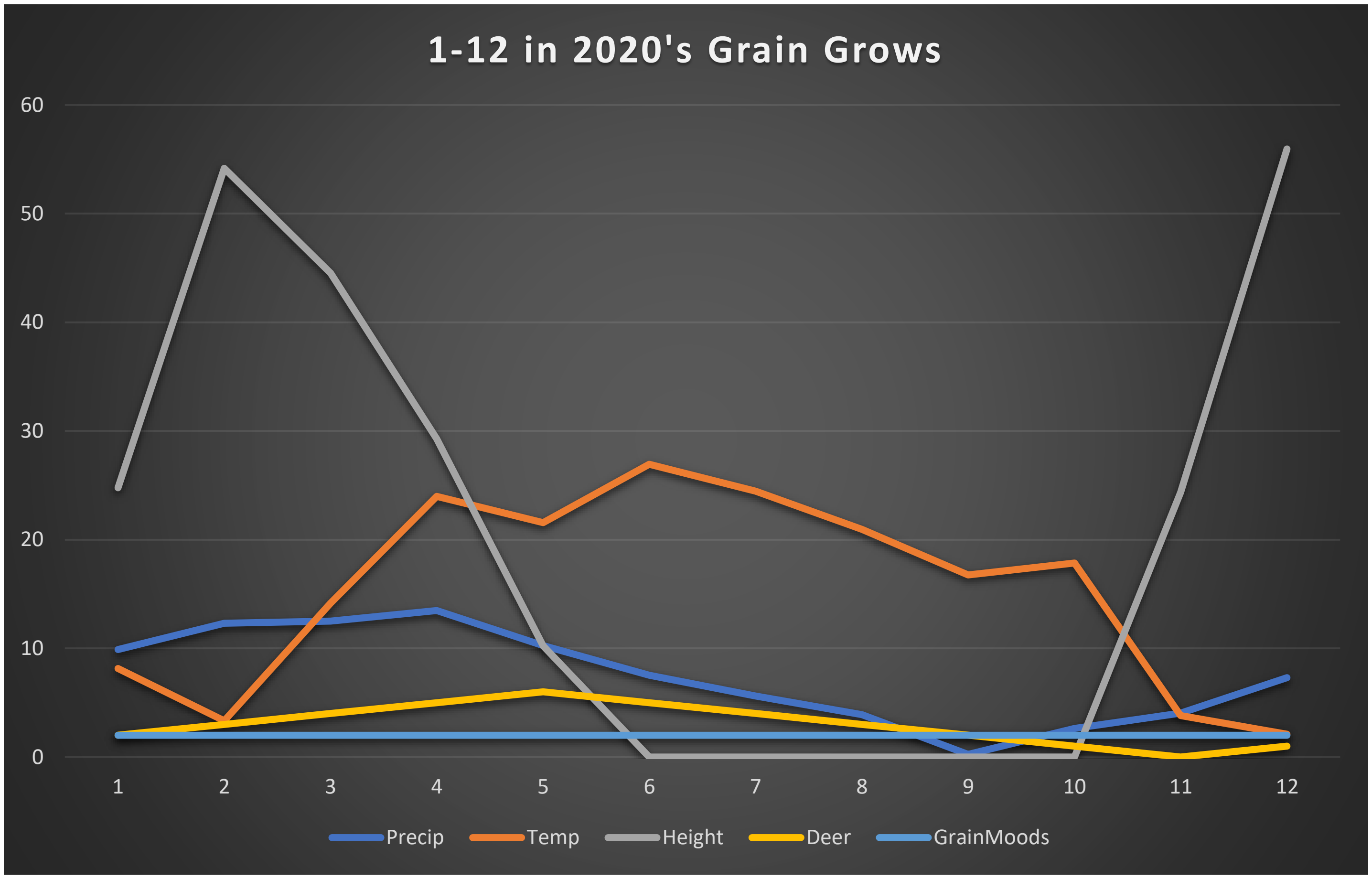
```
    float height = NowHeight;
    float tempFactor = exp(   -SQR(   ( NowTemp - MIDTEMP ) / 10.   )   );
    float precipFactor = exp(   -SQR(   ( NowPrecip - MIDPRECIP ) / 10.   )   );
    height += tempFactor * precipFactor * GRAIN_GROWS_PER_MONTH;


    if(GrainMoods >=6)
        height += 2;
    else if(GrainMoods >=3)
        height += 1;
    else
        height += 0;

    height -= (float)NowNumDeer * ONE_DEER_EATS_PER_MONTH;
```

OK, let's take a look at Graph.



Graph data is based on the table above. Using Temperature(°C) and Grain's Height(cm).

Since there is a lot of data for six years and 72 months, we choose one of the data for 2020 to analyze.



From the above graph, we can see that the growth of cereals is indeed affected by temperature, precipitation, deers, and GrainMoods. From June to October, the number of deer was too large, resulting in no grain at all, and the deer's data decreased rapidly after that. However, deer herds have a more significant impact on the growth of cereals. It may be because the growth pattern of the deer herd is linear, but the actual situation is normally distributed. In short, this project is still interesting.

```c
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <omp.h>


//Set the number of threads
#ifndef NUMT
#define NUMT        4
#endif


unsigned int seed = 0;


//"state" of the system
int NowYear;        // 2020 - 2025
int NowMonth;       // 0 - 11

float   NowPrecip;      // inches of rain per month
float   NowTemp;        // temperature this month
float   NowHeight;      // grain height in inches
int     NowNumDeer;     // number of deer in the current population

int  GrainMoods;  //Grain is in a good mood and grows fast


// Your basic time step will be one month
const float GRAIN_GROWS_PER_MONTH =         16.0; //inches
const float ONE_DEER_EATS_PER_MONTH =       1.5;

const float AVG_PRECIP_PER_MONTH =      7.0;    // average
const float AMP_PRECIP_PER_MONTH =      6.0;    // plus or minus
const float RANDOM_PRECIP =         2.0;        // plus or minus noise

const float AVG_TEMP =              60.0;      // average  Units of temperature a
re degrees Fahrenheit
const float AMP_TEMP =              20.0;      // plus or minus
const float RANDOM_TEMP =           10.0;      // plus or minus noise

const float MIDTEMP =               40.0;
const float MIDPRECIP =             10.0;


float Ranf( unsigned int *seedp,  float low, float high );
int Ranf( unsigned int *seedp, int ilow, int ihigh );
```

```
float SQR( float x );


void GrainDeer();
void Watcher();
void Grain();
void GrainMood(); // MyAgent  Grain is in a good mood and grows fast, but in a ba
d mood and grows slowly,lol


int main( int argc, char **argv )
{

    //Verify OpenMP support
    #ifndef _OPENMP
        fprintf( stderr, "No OpenMP support!\n" );
        return 1;
    #endif

    //Random function
    float x = Ranf( &seed, -1.f, 1.f );

    // starting date and time:
    NowMonth =    0;
    NowYear  = 2020;

    // starting state (feel free to change this if you want):
    NowNumDeer = 1;
    NowHeight =  1.;


    // The temperature and precipitation are a function of the particular month
    float ang = (  30.*(float)NowMonth + 15.   ) * ( M_PI / 180. );

    float temp = AVG_TEMP - AMP_TEMP * cos( ang );
    NowTemp = temp + Ranf( &seed, -RANDOM_TEMP, RANDOM_TEMP );

    float precip = AVG_PRECIP_PER_MONTH + AMP_PRECIP_PER_MONTH * sin( ang );
    NowPrecip = precip + Ranf( &seed,  -RANDOM_PRECIP, RANDOM_PRECIP );

    if( NowPrecip < 0. )
            NowPrecip = 0.;
    omp_set_num_threads(NUMT);
    #pragma omp parallel sections
    {
        #pragma omp section
        {
            Watcher();

        }
        #pragma omp section
        {
            Grain();
        }
        #pragma omp section
        {
            GrainDeer();
        }
        #pragma omp section
        {
            GrainMood();
        }
    }

    // implied barrier -- all functions must return in order
    // to allow any of them to get past here
    return 0;
}



//help function
float Ranf(unsigned int *seedp, float low, float high) {
    float r = (float) rand_r(seedp);
    return (low + r * (high - low) / (float)RAND_MAX);
}

int Ranf(unsigned int *seedp, int ilow, int ihigh) {
    float low = (float)ilow;
    float high = (float)ihigh + 0.9999f;
    return (int)(Ranf(seedp, low, high));
}

float SQR(float x) {
    return x * x;
}
```

```
void GrainDeer()
{

    while( NowYear < 2026 )
    {
    // compute a temporary next-value for this quantity
    // based on the current state of the simulation:
    float height = NowHeight;
    int deer = NowNumDeer;

    if (deer > height)
      deer--;
    else
        deer ++ ;

    // too mant deer
        //deer -= deer/1.1;

    if(deer < 0)
        deer =0;

    // DoneComputing barrier:
    #pragma omp barrier
    NowNumDeer = deer;

    // DoneAssigning barrier:
    #pragma omp barrier


    // DonePrinting barrier:
    #pragma omp barrier

    }



}

void Watcher()
{

    while( NowYear < 2026 )
    {
    // compute a temporary next-value for this quantity
    // based on the current state of the simulation:


    // DoneComputing barrier:
    #pragma omp barrier


    // DoneAssigning barrier:
    #pragma omp barrier
    //printf to csv
    //NowHeight *= 2.54; //inch ->to cm
    //NowTemp =(5./9.)*(NowTemp-32); //F -> oC
    printf("%d-
%d\t%6.2f\t%6.2f\t%6.2f\t%d\t%d\n", NowYear,NowMonth+1, NowPrecip, (5./9.)*(NowTe
mp-32), NowHeight*2.54, NowNumDeer, GrainMoods);



    // next year or next month
    if (NowMonth == 11) {
      NowMonth = 0;
      NowYear++;
    }
    else
      NowMonth++;

    //The temperature and precipitation are a function of the particular month
    //copy from http://web.engr.oregonstate.edu/~mjb/cs575/Projects/proj03.html
    float ang = (  30.*(float)NowMonth + 15.   ) * ( M_PI / 180. );

    float temp = AVG_TEMP - AMP_TEMP * cos( ang );
    NowTemp = temp + Ranf( &seed, -RANDOM_TEMP, RANDOM_TEMP );

    float precip = AVG_PRECIP_PER_MONTH + AMP_PRECIP_PER_MONTH * sin( ang );
    NowPrecip = precip + Ranf( &seed,  -RANDOM_PRECIP, RANDOM_PRECIP );

    if( NowPrecip < 0. )
            NowPrecip = 0.;

    // DonePrinting barrier:
    #pragma omp barrier

    }
```

```
}


void Grain()
{

    while( NowYear < 2026 )
    {
    // compute a temporary next-value for this quantity
    // based on the current state of the simulation:
    float height = NowHeight;
    //copy from http://web.engr.oregonstate.edu/~mjb/cs575/Projects/proj03.html
    float tempFactor = exp(   -SQR(   ( NowTemp - MIDTEMP ) / 10.   )   );
    float precipFactor = exp(   -SQR(   ( NowPrecip - MIDPRECIP ) / 10.   )   );

    //Be sure to clamp nextHeight against zero.
    height += tempFactor * precipFactor * GRAIN_GROWS_PER_MONTH;



    //test if happy, trying
    if(GrainMoods >=6)
        height += 2;
    else if(GrainMoods >=3)
        height += 1;
    else
        height += 0;


    height -= (float)NowNumDeer * ONE_DEER_EATS_PER_MONTH;


    if(height<0)
        height = 0;


    // DoneComputing barrier:
    #pragma omp barrier
    NowHeight = height;

    // DoneAssigning barrier:
    #pragma omp barrier
    // DonePrinting barrier:
    #pragma omp barrier

    }



}


// MyAgent  Grain is in a good mood and grows fast, but in a bad mood and grows slowly,lol

void GrainMood()
{

    while( NowYear < 2026 )
    {
    // compute a temporary next-value for this quantity
    // based on the current state of the simulation:
    int mood = GrainMoods;

    mood = 1;
    //Calculate mood index

    if(NowYear % 400 == 0 || (NowYear % 4 == 0 && NowYear % 100 != 0))
        mood += 1;
    else if(NowTemp >= 42.0 && NowTemp <=62.0 )
        mood +=3;
    else if(NowPrecip > 5)
        mood += 2;
    else if(NowMonth == 5 || NowMonth == 6)
        mood += 2;
    else if (NowMonth == 11 &&NowTemp <= 50 )
        mood -=1 ;
    else
        mood -= 2;


    if (mood <= 0)
         mood = 0;

    // DoneComputing barrier:
```

```
    #pragma omp barrier

    GrainMoods = mood;

    // DoneAssigning barrier:
    #pragma omp barrier


    // DonePrinting barrier:
    #pragma omp barrier


    }



}
```