## ITCS 4145/5145 Parallel Programming Final exam 5:00 pm to 7:30 pm, Tuesday May 6th, 2014

	Name:	
This is	s a closed book test. Do not refer to any materials except those supplied for the test. Answer questions in provided below questions. Use additional paper if necessary but make sure your name is on every sheet.	
	ied: "Basic MPI routines" and "Summary of OpenMP 3.0 C/C++ Syntax".  7 pag Total	50
Qu. 1	Answer each of the following briefly:	30)
a)	Why does Amdahl's law suggest that parallel computing has limited speedup? What did Gustafson suggest to counter Amdahl's argument?	2
b)	Suggest one reason for using a pattern programming approach.	2
c)	In the Seeds framework, typically the programmer has one method defined in the module class that is called in the Bootstrap class (with an instance of the module)? Suggest what that method might do.	2

d) What is the command to execute MPI program on the UNCC cluster cci-gridgw.uncc.edu?

2

e) In the routine MPI\_Send(message,13,MPICHAR,x,10, MPI\_COMM\_WORLD), when can x be altered without affecting the message being transferred?

2

f) A good implementation of the MPI routine MPI\_Barrier() should be re-entrant. Why?

2

g) Briefly what is the difference between a process and a thread?

2

h) Suppose two concurrent threads execute the code sequence:

$$y = 1;$$

$$x = y + 1;$$

where x and y are shared variables. What are the possible values for x and y after both threads have completed. Clearly show how you got your answer.

i) What does the Paraguin compiler do?

2

j) What does the Paraguin directive #pragma paraguin begin\_parallel do?

2

k) In OpenMP, several directives expect a structured block after the directive. Why is it not allowed to start that structured block with a { on the same line as the directive?

1) What does the following do?

for all 
$$(i = 1; i \le 10; i++) A[i] = B[i-1];$$

2

m) In Assignment 4 (Suzaku framework), what is the suzaku.o file?

n)	Show how to d	o matrix-vector	multiplication	using	a pipeline	pattern.
----	---------------	-----------------	----------------	-------	------------	----------

2

Using the CUDA equations: 0)

```
int x = threadIdx.x + blockIdx.x*blockDim.x;
int y = threadldx.y + blockldx.y*blockDim.y;
int index = x + y * N;
```

for flattening 2-dimensional addressing (x, y) into a 1 dimension index as a basis, develop an equation to flatten a 3-dimensional addressing (x, y, z) into a 1 dimension index. (There is more than solution.)

Qu. 2 Sequential code for odd-even transposition sort is given below:

```
int temp, i, j;
                                             // N steps with N numbers
for (i = 0; i < N; i++){
                                             // odd step, i odd
  if (i \& 1 == 1){
                                             // for each even j index
      for (j = 2; j < N; j+=2)
                                             // compare number with one before it
        if (a[j] < a[j-1]) {
            temp = a[j-1];
                                             // swap elements
            a[j-1] = a[j]
            a]j] = temp;
                                             // even step, i even
   } else {
                                             // for each odd j index
      for (j = 1; j < N; j+=2)
                                             // compare number with one before it
         if (a[j] < a[j-1]) {
                                             // swap elements
            temp = a[j-1];
            a[j-1] = a[j]
            a]j] = temp;
   }
}
```

(If there are any mistakes in this code, correct them.)

(a) Show how the code sorts the numbers 4 2 7 8 5 1 3 6.

In the following, provide very clear explanation of how your programs work. If I do not understand the code, I will assume it is incorrect. It is not necessary to give complete program, just that part that performs the sorting.

(b) Write an OpenMP program to implement odd-even transposition sort using N/2 threads (N even).

(c) Write an MPI program to implement odd-even transposition sort assuming one number is stored in each process (N processes). (It may be easier to have each pair of processes send their numbers and both do the comparison.)