

**ITCS 4/5145 Parallel Computing**  
**Test 1**  
**5:00 pm - 6:15 pm, Thursday Feb 20, 2014**

Name: .....

This is a closed book test. Do not refer to any materials except those supplied for the test.

Supplied: “*Basic MPI routines*” and “*Summary of OpenMP 3.0 C/C++ Syntax.*”

Answer questions in space provided below questions. Use additional paper if necessary but make sure your name is on additional sheets.

Total /40  
6 pages

Qu. 1 Answer each of the following briefly:

- (a) What is the maximum speed-up of a parallel computation given that 90% of the computation can be divided into six equal parts that can be executed at the same time?  
*Clearly show how you obtained your answer. (No points for simply writing a numeric answer without showing how you got the answer, even if correct.)*

2

- (b) Give one advantage for using parallel patterns.

2

- (c) Give one disadvantage for using patterns.

2

- (d) In the Seeds framework a parameter called “segment” appears in the DiffuseData and GatherData methods. How is this parameter used for matrix addition? 2
- (e) There are three versions of the Java based Seeds Framework. Which did you use for Assignment 1? 2
- (f) When does the MPI routine MPI\_Send() return? 2
- (g) In Assignment 2, which command did you use to compare the output of a sequential C program with that of a MPI program? 2
- (h) In Assignment 2, how did you specify the computers to use on the UNC-C cluster? 2

(i) What is the fundamental difference between a process and a thread? 2

(j) What is a detached thread? Suggest how one might create a detached thread in OpenMP.  
(Not said specifically in lecture notes) 2

(k) What is a critical section? 2

(l) In OpenMP, how can one make five threads do completely different code sequences? 2

In the following, provide comments in your code to help the grader. *If I do not understand the code, I will assume it is incorrect.* The programs should be *complete programs* of the form:

```
#include <stdio.h>
#include <stdlib.h>
#include "mpi.h"
#include <omp.h>
main(int argc, char **argv ) {
    ...
    return(0)
}
```

You are to have all required statements for a working program.

Supplied: “*Basic MPI routines*” and “*Summary of OpenMP 3.0 C/C++ Syntax.*”

Qu. 2 (a) First write a sequential C program to compare two integer  $N \times N$  arrays and report how many elements are different (i.e. if  $A[i][j]$  is different to  $B[i][j]$ , for all  $i$  and  $j$ ) with a print statement.  $N$  is a defined constant set to 1000. You can assume the arrays are initialized with values but show where that would be in the program with comments.

(b) Modify the code using OpenMP directives to parallelize the code using  $P$  threads, where  $P$  is also a defined constant and  $N$  is a multiple of  $P$ . Write this code below.

6

(c) Finally re-write the code to be an MPI program with  $P$  processes.