Teleclass - Introduction Parallel Programming
Test 2
Thursday November 17th, 1998, 8:00 am to 9:20 am

Three pages.

Attempt all questions in the spaces provided. **Name:** ....

Use additional paper if necessary.

You may refer to the attached Appendix A “Basic PVM routines”, and Appendix B “Basic MPI routines”. Do not refer to any other materials.

Mark/30

Qu. 1 Answer each of the following briefly:

(a) What is a *barrier* in the context of PVM and MPI routines? 2

(b) What is meant by the term *data parallel computation*? 2

(c) What is meant by the term *receiver-initiated* in decentralized dynamic load balancing? 2

(d) What is the difference between a (heavyweight) process and a thread? 2

(e) What is a detached thread? 2

(f) What is meant by the term *thread-safe*? 2

(g) What does the following code do: 2

```
forall (i = 0; i < n; i++) {
   a[i] = a[i + n];
}
```

(h) State Berstein's conditions for determining whether two statements can be executed in parallel. 2

Qu. 2 Show the steps in sorting the following sequence using odd-even transposition sort (a parallel variation of bubble sort): 4

4, 2, 7, 8, 5, 1, 3, 6

Qu. 3 Rank Sort code given below:

```
     for (i = 0; i < n; i++) {          /* for each number */
       x = 0;
       for (j = 0; j < n; j++)          /* count number of nos less than it
*/
          if (a[i] > a[j]) x++;
       b[x] = a[i];                     /* copy number into correct place */
     }
```

Write a parallel program (in PVM, MPI, or Pthreads) for Ranksort. It is not necessary maintain the order of identical numbers in the sorted list. <u>Give clear comments explaining the code.</u> 10