Name:______________________________

SS# ________________________________

CS 320
Introduction to Parallel Programming
Final Exam
Due: Tuesday May 12,1998, at 9:00 am
Bring exam to 3317 DCL

NOTE: Because this is a take-home exam, we cannot answer any questions. So, if there are any ambiguities, please state your assumptions.

1. [5 pts.] What is the effect of Moore's Law on the popularization of parallel computing?

2. [5pts.] Briefly discuss how supercomputers contribute to lower costs and enhance the quality of designs in the aerospace industry.

3. [5pts.] (a) Define locality in the context of multicomputers. Why is locality important?

   (b) Give two examples of SIMD machines.

4. [5pts.] (a) Mention one advantage of the array notation over the parallel loop notation. (b) Mention a disadvantage.

5. [5pts] What is a Fortran program unit?

6. [5 pts.] Under which circumstances is the following `where` statement *incorrect*? Assume that subscripts are always within the bounds of the array.

```
where (a(1:n+1) > b(m:m+n)) x(1:100)=y(1:100)+1
```

7. [10 pts.]Consider the code segment

```
...
t=a+b
y=c*d
t=t+y
do i=1,n
     x(i)=x(i)+t
end do
print *,x
end
.
```

   The following translation to parallel form is incorrect:

```
       ...
c$omp  parallel private(y)
       t=a+b
       y=c*d
       t=t+y
c$omp  pdo
       do i=1,n
            x(i)=x(i)+t
       end do
c$omp  end pdo nowait
c$omp  end parallel
       print *,x
       end
```

Explain why the code is incorrect. Then, give two different correct versions of the previous translation following the following rules: (a) do not remove any of the OpenMp directives, (b) do not change or delete any of the statements, and (c) do not change the location of any of the statements or OpenMp directives.

8. [10 pts.] Consider the code segment

```
c$omp  parallel
c$omp  pdo
       do i=1,n
            a(i)=b(i)+c(i)
            do while (a(i) > x(i))
                 a(i) = a(i) - x(i)
            end do
       end do
c$omp  end pdo nowait
c$omp  pdo
       do i=1,n
            c(i) = d(i) + e(i)
       end do
c$omp  end pdo nowait
c$omp  end parallel
```

(a) Explain why each of the `nowait` directives could improve performance.
(b) How would you change the second loop so that the first `nowait` causes the parallel program to contain races?

9. [5 pts.] Consider the code segment:

```
c$omp       parallel do
            do i=1,n
                 x(k(i))=x(k(i)+1)+B(i)
            end do
c$omp       end parallel
```

(a) Are races possible in this loop? Explain (b) Is the loop a valid translation of its sequential counterpart? Explain.

10. [5 pts.] Transform the following loop to parallel form:

```
do i=1,n
     k=2*k
     x=b(i)+c(i)
     a(k)=x+sin(x)
end do
```

11. [10 pts.] Consider the loop:

```
do i=3,n
     do j=2,n
          a(i,j)=a(i+2,j-1)+a(i-1,j)+a(i,j-1)
     end do
end do
```

(a) Sketch the dependence graph of the unrolled two dimensional iteration space. Clearly mark anti dependences in the graph.
(b)Transform the loop into a `forall` vector operation with the maximum parallelism possible.

12. [10 pts.]Assume a vector A is distributed across N processors, one element per processor. Using send/receive as the only communication primitives, write an MPI program that computes in processor i the value:

$$\sum_{j=1}^{i} A(j)$$

13. [10pts.] Transform the following loop into vector parallel form:

```
do i=1,n
     if (x(i).le.x(i-1)) x(i)=x(i-1)
end do
```

Is the resulting algorithm consistent? Explain.

14. [5pts.] Assuming that N is very large, and using only loop interchanging (if necessary), transform the following Fortran loop into the best possible parallel version. Justify your answer.

```
do i=1,n
     do j=1,n
          a(i,j)=a(i,j)+1
     end do
end do
```

15. [5pts.] Give a brief intuitive reason why asynchronous iterative algorithms may converge faster than their synchronous equivalents.