Lab 2

A. Write and run an OpenMP program that:

Using the skeleton in Lab2a.c, add statements to the program to:

1. Determines the number of processors available to run the program.
2. Prints out a unique threadId for each thread using an OpenMP built-in function
3. Determines which thread executes a *master* and *single* statement of a parallel region.

B. **Write and time 3 ways of doing reductions, two of which will be parallel. Initialize within the program a single-dimensioned array *a* with 1,000,000 elements.** You will need to add statements as indicated in the UPPER CASE COMMENTS. The program will:

a. Perform a sequential reduction on *a* into the variable *s*, that is, *s* should equal the sum of all of the elements of *a*.
b. Perform a reduction using a parallel for, and print its time

```
int nt = numberofthreads
int res[nt];
#pragma omp parallel for
 for (i=0; i < 1,000,000; i++) {
   res[mythread] += a[i];
}
```

c. use an OpenMP reduction to create the sum, and print its time

Think about why the times are different, and why they are different.

C. The program in Lab2c.c creates the sum of 1/(float) i, where i goes from 1 to 100000000, in the first loop, and from 100000000 to 1 in the second loop. Think about and write down what you think about the two results.