# LA Worksheet #9

1. The size_t variable is declared outside of the parallel sections, meaning the threads share the same iterator.

2.
```
int i,j;
#pragma omp parallel for private(i,j) shared
 (array,n)
 for(i=0; i<n; i++)
    for(j=1; j<n; j++) {
       array[i][j] += array[i][j-1];
    }
```
The 1-D case may cause race conditions.

3.
```
void hello(long *old, long* new, int n){
   int i;
   double sumWeights=0, sum=0
   sum = n* old[0];
 #pragma omp parallel for reduction(+:sumWeights)
   for(i=0; i<n; i++){
      new[i]= old[i] * exp(100.0f/old[i]);
      sumWeights += new[i]
   }
   sumWeights /= sum;
 #pragma omp parallel for
   for (i=0; i<n; i++)
      new[i]=new[i]/sumWeights;
 }
```

4. Dynamic linking occurs every time the program is run while static linking only occurs when it is compiled. Dynamic linking allows for libraries to be shared, but may be slower due to linking during runtime.

5. a) abort, synchronous, DNR
   b) interrupt, asynchronous, return to next instruction
   c) fault, synchronous, attempts to retrieve page from VM
   d) trap, synchronous, return to next instruction

6.

| | VPN | VPO | PPN | PPO |
|---|---|---|---|---|
| 1KB | 22 | 10 | 14 | 10 |
| 2KB | 21 | 11 | 13 | 11 |
| 4KB | 20 | 12 | 12 | 12 |
| 8KB | 19 | 13 | 11 | 13 |