1) Write a parallel program using OpenMP to perform vector addition, subtraction, multiplication. Demonstrate task level parallelism. Analyze the speedup and efficiency of the parallelized code.

2) Write a parallel program using OpenMP to find sum of N numbers using the following constructs/clauses.
   a. Critical section
   b. Atomic
   c. Reduction
   d. Master
   e. Locks

3) Write a parallel program using OpenMP to implement the Odd-even transposition sort. Vary the input size and analyse the program efficiency.

   Hint: Odd-even transposition sort is a sorting algorithm that's similar to bubble sort. The list **a** stores **n** integers, and the algorithm sorts them into increasing order. During an *"even phase"* (phase % 2 == 0), each odd-subscripted element, a[i], is compared to the element to its *"left"* a[i−1], and if they're out of order, they're swapped. During an *"odd"* phase, each odd-subscripted element is compared to the element to its right, and if they're out of order, they're swapped. A theorem guarantees that after n phases, the list will be sorted.

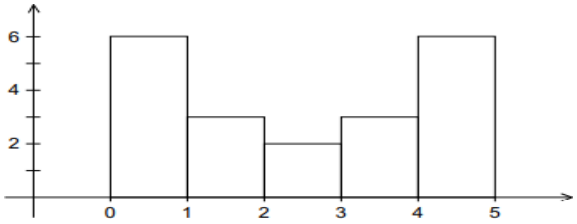
	Subscript in Array						
Phase	**0**		**1**		**2**		**3**
0	9	↔	7		8	↔	6
	7		9		6		8
1	7		9	↔	6		8
	7		6		9		8
2	7	↔	6		9	↔	8
	6		7		8		9
3	6		7	↔	8		9
	6		7		8		9

4) Write an OpenMP program to find the Summation of integers from a given interval. Analyze the performance of various iteration scheduling strategies.

5) Write a parallel program using OpenMP to generate the histogram of the given array A.

   Hint: To generate histogram, we simply divide the range of the data up into equal sized sub intervals, or bins and determine the number of measurements (frequency) in each bin.

Example: suppose our data are
1.3, 2.9, 0.4, 0.3, 1.3, 4.4, 1.7, 0.4, 3.2, 0.3, 4.9, 2.4, 3.1, 4.4, 3.9, 0.4, 4.2, 4.5, 4.9, 0.9.



Where, Y axis represents the frequency of occurrence of the values and the x axis represents the bins.