Assignment of OpenMP:

Problem 1

Parallelize Homework 1, the integration using Trapezoidal and Simpson Rules.

Choose a value of N, large enough so that there is several seconds running time. Using 1 thread, 2 threads, 4 threads and 8 threads and complete the following table:

My choice N = ______________

Thread Number	Total Time (sec.)	Speedup
1	T1 =	1
2	T2 =	T1/T2 =
4	T4 =	T1/T4 =
8	T8 =	T1/T8 =

Here the total time is the time for integration calculations.
If you know any plot package, you can plot the Speedup graph (optional).

As usual, provide a Makefile, document your various functions with docstirngs, and provide short report stating what you have along with your results.

We expect a linear speedup for two threads. Beyond two, the speedup might depend on the processor you use.

Problem 2

Parallelize the attached code, charfreq.cpp. All array sizes are specified in the code. The code considers an array of size *n* stores integers *m[i]* in the range [0 to 255], where *i* ranges from 0 to *n*-1. The code computes the fraction of array elements that store each index. The sum of the fractions is unity. The loop to parallelize is marked in the code. Time the loop as it is currently, and time the loop running with 2, 4, and 8 threads. [hint: it is more tricky than it appears at first glance. Use loop scheduling to accelerate the code]. Time each thread and print the timings in your report. Explain why the code runs faster with your chosen implementation.