1. Explain what OpenMP is, what are it’s benefits?

OpenMP (Open Multi-processing) is an API for writing multithreaded applications. It contains compiler directives and libraries that help write parallel applications.

* Good performance and scalability
* An OpenMP program is portable
* Requires little programming effort
* Allows program to be parallelized incrementally

1. What are the #pragma definitions, what do they do?

The ' #pragma ' directive is the method specified by the C standard for providing additional information to the compiler, beyond what is conveyed in the language itself. The forms of this directive (commonly known as pragmas) specified by C standard are prefixed with STDC.

1. Write the OpenMP #pragma definition to execute a loop in parallel.

#pragma omp parallel

#pragma omp for

For (i=0; i<N; I++){ Perform something}

1. What does the reduction do in the #pragma definition in OpenMP?

Reduction is used when we have variables that multiple threads are changing throughout the parallel loop. To handle this, we use reductuon(operation:var), where operation is how we want to deal with the variables and var is the variable name.

1. Explain the critical and private declarations used in OpenMP.

The private Declaration specifies variables which a parallel section should allocate for each individual thread. Each thread has its own instances of the variables as private, The Critical declaration defines the critical section in a parallel section of code, it allows 1 thread execution at a time