Dated :
Assessment No. : 1

Aim:

Write a simple OpenMP program to demonstrate the parallel loop construct.

a. Use OMP_SET_THREAD_NUM( ) and OMP_GET_THREAD_NUM( ) to find the number of processing unit
b. Use function invoke to print 'Hello World'
c. To examine the above scenario, the functions such as omp_get_num_procs(), omp_set_num_threads(), omp_get_num_threads(), omp_in_parallel(), omp_get_dynamic() and omp_get_nested() are listed and the explanation is given below to explore the concept practically.

   omp_set_num_threads() - takes an integer argument and requests that the Operating System provide that number of threads in subsequent parallel regions.

   omp_get_num_threads() (integer function) - returns the actual number of threads in the current team of threads.

   omp_get_thread_num() (integer function) - returns the ID of a thread, where the ID ranges from 0 to the number of threads minus 1. The thread with the ID of 0 is the master thread.

   omp_get_num_procs() - returns the number of processors that are available when the function is called.

   omp_get_dynamic() - returns a value that indicates if the number of threads available in subsequent parallel region can be adjusted by the run time. o omp_get_nested( ) returns a value that indicates if nested parallelism is enabled.

SOURCE CODE:

EXECUTION:

REMARKS: