

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:**