Walchand College of Engineering, Sangli
Department of Computer Science and Engineering
Final Year: High Performance Computing Lab 2022-23 Sem I
Class: Final Year (Computer Science and Engineering)
Year: 2022-23 Semester: 1
Course: High Performance Computing Lab
Assignment No : 1

Q1. Write Program for Hello World by making use of openmp

A_01 > 1_hello_world_ll.cpp > ...

```cpp
#include<omp.h>
#include<bits/stdc++.h>
int main(){

    #pragma omp parallel
    {
        printf("Hello World from process: %d\n", omp_get_thread_num());
    }
    return 0;
}
```

```
pavan7494@pavan7494:~/Desktop/pavan7494/Other/CP/HPC_LAB/A_01$ g++ -fopenmp 1_hello_world_ll.cpp
pavan7494@pavan7494:~/Desktop/pavan7494/Other/CP/HPC_LAB/A_01$ ./a.out
Hello World from process: 0
Hello World from process: 2
Hello World from process: 3
Hello World from process: 6
Hello World from process: 5
Hello World from process: 1
Hello World from process: 7
Hello World from process: 4
pavan7494@pavan7494:~/Desktop/pavan7494/Other/CP/HPC_LAB/A_01$
```

Information :

When the master thread reaches this line, it forks additional threads to carry out the work enclosed in the block following the #pragma construct. The block is executed by all threads in parallel. The original thread will be denoted as master thread with thread-id 0.Once the compiler encounters the parallel regions code, the master thread (thread which has thread id 0) will fork into the specified

number of threads. Entire code within the parallel region will be executed by all threads concurrently. Once the parallel region ended, all threads will get merged into the master thread.

Q2. Get sum of square of first n natural num using openmp

A_01 > 1_Squares_l.cpp > ...

```cpp

#include<omp.h>
#include<bits/stdc++.h>
int main(){
    int sum=0;
    #pragma omp parallel for
    for(int i=1; i<=10;i++)
    {
        printf("thread No. %d Number : %d Square : %d\n", omp_get_thread_num(), i, i * i);
        sum+=i*i;
        printf("Sum is %d \n",sum);
    }
    printf("%d",sum);
    return 0;

}


```

```
pavan7494@pavan7494:~/Desktop/pavan7494/Other/CP/HPC_LAB/A_01$ g++ -fopenmp 1_Squares_1.cpp
pavan7494@pavan7494:~/Desktop/pavan7494/Other/CP/HPC_LAB/A_01$ ./a.out
thread No. 7 Number : 10 Square : 100
Sum is 100
thread No. 0 Number : 1 Square : 1
Sum is 101
thread No. 0 Number : 2 Square : 4
Sum is 105
thread No. 5 Number : 8 Square : 64
Sum is 169
thread No. 2 Number : 5 Square : 25
Sum is 194
thread No. 1 Number : 3 Square : 9
Sum is 203
thread No. 1 Number : 4 Square : 16
Sum is 219
thread No. 4 Number : 7 Square : 49
Sum is 268
thread No. 3 Number : 6 Square : 36
Sum is 304
thread No. 6 Number : 9 Square : 81
Sum is 385
385pavan7494@pavan7494:~/Desktop/pavan7494/Other/CP/HPC_LAB/A_01$
```

Information 2:

Once the compiler encounters the parallel regions code, threads will get created and it will calculate square of all first 100 numbers with addition of all squares of numbers stored in variable sum. Threads works in parallel and do task given to it.