**Example program: 1**

//compute the sum of two arrays in parallel

#include < stdio.h >

#include < omp.h >

#define N 1000000

int main(void) {

float a[N], b[N], c[N];

int i;

/\* Initialize arrays a and b \*/

for (i = 0; i < N; i++) {

a[i] = i \* 2.0;

b[i] = i \* 3.0;

}

/\* Compute values of array c = a+b in parallel. \*/

#pragma omp parallel shared(a, b, c) private(i)

{

#pragma omp for

for (i = 0; i < N; i++) {

c[i] = a[i] + b[i];

printf ("%f\n", c[10]);

}

}

}

**Example program 2:**

Add all elements in an array in parallel

#include < stdio.h >

int main() {

const int N=100;

int a[N];

//initialize

for (int i=0; i < N; i++)

a[i] = i;

//compute sum

int local\_sum, sum;

#pragma omp parallel private(local\_sum) shared(sum)

{

local\_sum =0;

//the array is distributde statically between threads

#pragma omp for schedule(static,1)

for (int i=0; i< N; i++) {

local\_sum += a[i];

}

//each thread calculated its local\_sum. ALl threads have to add to

//the global sum. It is critical that this operation is atomic.

#pragma omp critical

sum += local\_sum;

}

printf("sum=%d should be %d\n", sum, N\*(N-1)/2);

}

**Exercise for 25/07/2019**

1. Write a OpenMP program to find the count of prime numbers from the given input using appropriate constraint
2. Write a OpenMP program to find factorial of a 3 different numbers in separate code block. The output of all 3 factorial values should share a same variable through the critical section constraint.