

OpenMP

OpenMP Questions

1. (a) What is the maximum speedup possible in the following code:

```
#pragma omp parallel private(i,j)
{
    #pragma omp sections nowait
    { #pragma omp section
    for (i = 0; i < n ; i ++ )
        for(j =0 ; j<=i ; j++)
            b[j][i] = a[j][j] + a[j][i-1]/2
```

```
        #pragma omp section
        for (i = 1; i < m ; i ++ )
            for(j =0 ; j<=i ; j++)
                d[j][i] = c[j][j] + c[j][i-1]/2
    }
}
```

Ans: maximum speedup is 2, since there are only two sections. This means that the code cannot be paralleled on more than 2 threads.

1(b) How can you increase the speedup in the code above?

Ans: By parallelising the for loops using **#pragma omp for**

2. What is wrong with the code below:

```
#pragma omp parallel
{
    #pragma omp critical
    {
```

```
sum += c; // assume c is a constant
#pragma omp barrier
a += sum;
}
}
```

Ans: The code will deadlock as the thread in the critical section will block on `#pragma omp barrier` and no other threads will be able to execute the barrier instruction.

3. Print the o/p of the following code:

```
#pragma omp parallel num_threads(2)
{
printf(" Level 1: tid %d — num of threads %d \n", omp_get_thread_num(),
omp_get_num_threads());
#pragma omp parallel num_threads(3)
printf(" Level 2: tid %d —num of threads %d \n", omp_get_thread_num(), omp_get_num_threads());
}
```

Ans: Level 1: tid 0 — num of thread 2

Level 1: tid 1 — num of thread 2

Level 2: tid 0 — num of threads 3

Level 2: tid 1 — num of threads 3

Level 2: tid 2— num of threads 3
