

## Evaluation of Students Understanding of the Learning Objectives

### •**APPLY** basic OpenMP pragmas to accelerate a common scientific pattern

Sample Questions:

1. Rewrite provided code provided so instead of using only the neighbor to the North, South, East, and West uses also the neighbor

NW	North	NE
West	Center	East
SW	South	SE

2. Ask students to change the Heat parameters like SPEED, Increase/decrease the number of iterations for the stencil, and make them explain the difference in results

### •**USE** Profiler tools to evaluate bottlenecks in code

1. Use openMP timing functions to measure execution times of different loops and/or functions on the code. Example

```
double start;  
  
double end;  
  
start = omp_get_wtime();  
... work to be timed ...  
  
end = omp_get_wtime();  
  
printf(~Work took %f seconds\n~, end - start);
```

### •**IDENTIFY** which loops/task can be distributed with OpenMP pragmas(loop dependency)

1. Does the next loop contain a loop dependency?

```
for( i = 1; i<100; i++){  
    A(i) = B(i)+1  
    C(i) = A(i-1)*2    <- yes here, to calculate this iteration we need the prev.  
one
```

}

2. Does the next loop contain a loop dependency?

```
for( i = 1; i<100; i++){
```

```
    C(i) = A(i-1)*2    <- yes here, to calculate this iteration we need the prev.
    one
```

```
}
```

3. Does the next loop contain a loop dependency?

```
for( i = 1; i<100; i++){
```

```
    A(i) = B(i-1)*2    <- no
```

```
}
```

•**COMPARE** sequential to distributed execution times

1. Ask students to change the number of OpenMP threads used and obtain runtimes of sequential code (also provided) vs. distributed code using those different process

Create a table and graphs to study execution

Sequential	Execution time
OpenMP 1 Thread	
OpenMP 2 Threads	
....	

Possible more Advanced Questions for Students:

- (code) The stencil code :
- Rewrite code for stencil so instead of hardcoding in the loop the neighbor cells, make the code more flexible and take this neighborhood and its coefficients as a filter passed in one input parameter

- (code) Change it to 3D stencil code
- If a stencil or convolutional filter is separable it can be applied first to rows and then to the columns. Which one may be faster for acceleration ?
- On the `pragma omp for schedule(static,4)` what does it mean? Useful in this example?
- Will a static, 8 work better?

Ask students to run code and modify:

- Enhance memory allocation using: `mmap`
- Change stencil to 9 points instead of just 4
- Explore `Collapse()` clause
- Explore the `simd Aligned()` clause