

OpenMP

1. Write a parallel program implementing a bubble sort with OpenMP directives.

Bubble sort is a simple sorting algorithm that repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order. The pass through the list is repeated until the list is sorted. Let's say we have an array of numbers [5, 2, 9, 1, 6] and we want to sort it using the bubble sort algorithm.

- Compare the first two elements (5 and 2). Since 5 is greater than 2, we swap them: [2,5,9,1,6]
- Compare the next two elements (5 and 9). Since they are in the correct order, we move on.
- Compare the next two elements (9 & 1). Since 9 is greater than 1, we swap them: [2,5,1,9,6]
- Compare the next two elements (9 & 6). Since 9 is greater than 6, we swap them: [2,5,1,6,9]
- Repeat the above steps until no more swaps are needed.

Question 1. Can this algorithm be made parallel?

Question 2. Write a parallel program implementing this algorithm with OpenMP directives.

2. Write a parallel program that returns the maximum element in a vector of N elements

- Using « partial » maximums (each thread determines the maximum of a subset of the vector elements)
- Using a unique shared maximum, without a reduction clause
- Using the appropriate reduction clause

3. Write a parallel program using OpenMP that counts the number of prime numbers between 1 and N

A sequential version could be:

```
int num_primes = 0;
for (int i = 1 ; i<N ;i++){
bool is_prime = TRUE ;
for (int j=2 ; j<i ; j++){
if (i%j == 0){ is_prime = FALSE ; break ;}
} if (is_prime) num_primes++ ;
}
```