

Implementar los siguientes ejercicios a partir del ejemplo de cálculo del módulo de un vector con el diseño master/worker. Entregar una solución con una aplicación de consola, que invoque los métodos pedidos y muestre el resultado de los cálculos realizados.

1. (3 puntos) Modificar el proyecto de modo que calcule la suma vectorial de dos vectores de `int`, añadiendo dos atributos más de tipo vector, representando tanto los datos como el resultado. Realizar las modificaciones oportunas. Modificar el particionado de datos del siguiente modo:

Cada worker se crea con un atributo de tipo `int`: `thread`. Por ejemplo, si se crean cuatro los valores de ese atributo para cada thread serán 0, 1, 2, 3. Otro también `int`: `nThread`, representando el número de workers, en este ejemplo 4. Cada worker procesa los datos en las posiciones `thread`, `thread+nThread`, `thread+2*nThread`, etc. Entonces, si se usan 4 threads, el thread 0 procesa los elementos en 0,4,8,12,... el 1 en 1,5,9,13,... el 2 en 2,6,10,14,... el 3 en 3,7,11,15,...

2. (2.5 puntos) Usando `Parallel.For`, implementar un método reciba un vector de `int` y un `int`, devuelve la posición de la primera ocurrencia del valor en el vector o -1 si no está.
3. (2.25 puntos) Calcular la suma vectorial de dos vectores usando la siguiente estrategia: con `Invoke` lanzar dos tareas en paralelo de modo que una de ellas procese las posiciones pares y otra las impares.
4. (2.25 puntos) Calcular la suma de los cuadrados de los elementos de valor impar de un vector, usando `Parallel.For` o `Parallel.ForEach` más el método adecuado de `Interlocked`.