

```
// OMPTrial.cpp : Este archivo contiene la función "main". La ejecución del programa comienza y termina ahí.
//
```

```
#include <iostream>
#include <ctime>
#ifdef _OPENMP
#include <omp.h>
#else
#define omp_get_thread_num() 0
#endif
// en este primer paso definimos las librerias a usar al igual que verificamos si necesitaremos llamar OPENMP o no
//
```

```
#define N 1000
#define chunk 10
#define mostrar 5
```

```
// definimos el tamaño de muestra y el tamaño de división que usaremos
//
```

```
void imprimirArreglos(int* a, int* b, int* c) // esta parte definimos una función para mandar imprimir el elemento N de cada arreglo
```

```
{
    for (int i = 0; i < mostrar; i++)
    {
        std::cout << "A[" << i << "] = " << a[i];
        std::cout << "\n";
        std::cout << "B[" << i << "] = " << b[i];
        std::cout << "\n";
        std::cout << "La suma Final es C[" << i << "] = " << c[i];
        std::cout << "\n";
    }
}
```

```
void Suma(int* a, int* b, int* c) // definimos la función suma para tener un main más limpio
```

```
{
    for (int i = 0; i < N; ++i) // este es el For que tendremos que nos funcionara para el paralelismo
    {
        c[i] = a[i] + b[i];
    }
}
```

```
const int tamano = N+1, MAXIMO = 100, MAXIMO2=100; // definimos las constantes
```

que utilizaremos en el main

```
int main()
{
    int a[tamano] = { 0 }; //definimos la matriz A
    int i;
    for (i = 0; i < tamano; ++i)
        a[i] = rand() % MAXIMO;
    std::cout << "TERmine arreglo A:\n";
    int b[tamano] = { 0 }; // definimos la matriz B
    for (i = 0; i < tamano; ++i)
        b[i] = rand() % MAXIMO2;
    std::cout << "Terminar Arreglo B:\n";

    omp_set_num_threads(10);
    int c[tamano] = { 0 }; // definimos la matriz C
    Suma(a, b, c); // aplicamos la suma previamente definida
    imprimirArreglos(a, b, c); //imprimimos los arreglos previamente definidos
}
```