

MATERIA: COMPUTO EN LA NUBE

ALUMNO: FRANCISCO MEDELLIN ZERTUCHE

MATRICULA: A01794044

PROFESOR TITULAR: EDUARDO ANTONIO CENDEJAS CASTRO

TAREA: 1. Programación de una solución paralela

1. Introducción:

En las ciencias computacionales el concepto de paralelismo hace referencia a técnicas computacionales basadas en un principio “dividir un gran problema en varios y resolverlos al mismo tiempo”. En otras palabras, es el uso simultaneo de múltiples recursos computacionales para resolver un problema.

En este ejercicio se llevó acabo el diseño e implementación de un algoritmo paralelo mediante el uso del lenguaje de programación C++ y la librería OpenMP. El objetivo principal del algoritmo es la suma paralela entre arreglos.

2. Liga de Repositorio:

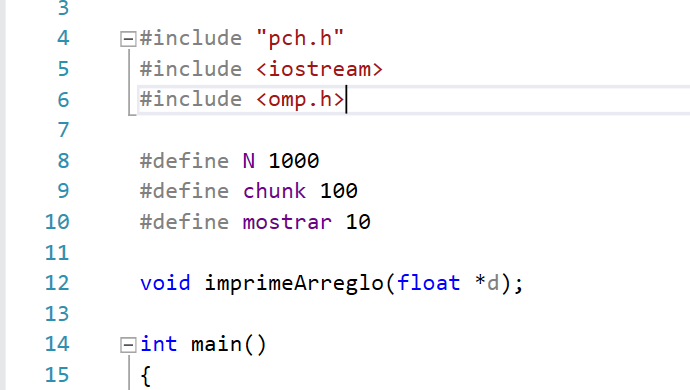
Dentro del repositorio para la materia cloud computing, se encuentra la carpeta “M2 Tarea1 ProgramacionParalela” donde se encuentran los archivos de esta actividad.

[Link de la carpeta de tareas en el repositorio de clase.](https://github.com/FranciscoMedellin/itsm-cloud-computing/tree/main/Tareas/M2%20Tarea1%20ProgramacionParalela)

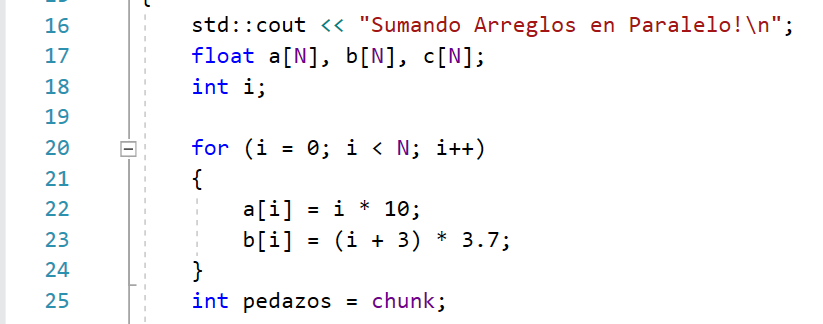
3. Ejecuciones del proyecto y explicación del código y resultados:

**Explicación del código:**

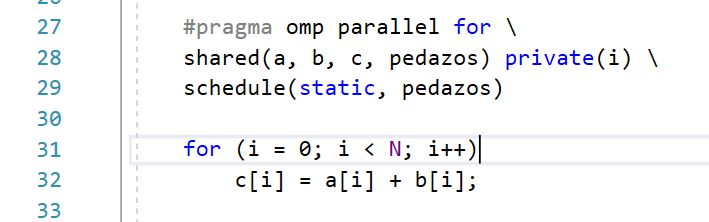
* Primero declaramos las librerías necesarias.
* Definimos las constantes **N** (cantidad de elementos a manejar en los arreglos), **chunk** (tamaño que tendrán los grupos de los arreglos para que cada hilo los opere), **mostrar** (Cantidad de datos a imprimir).



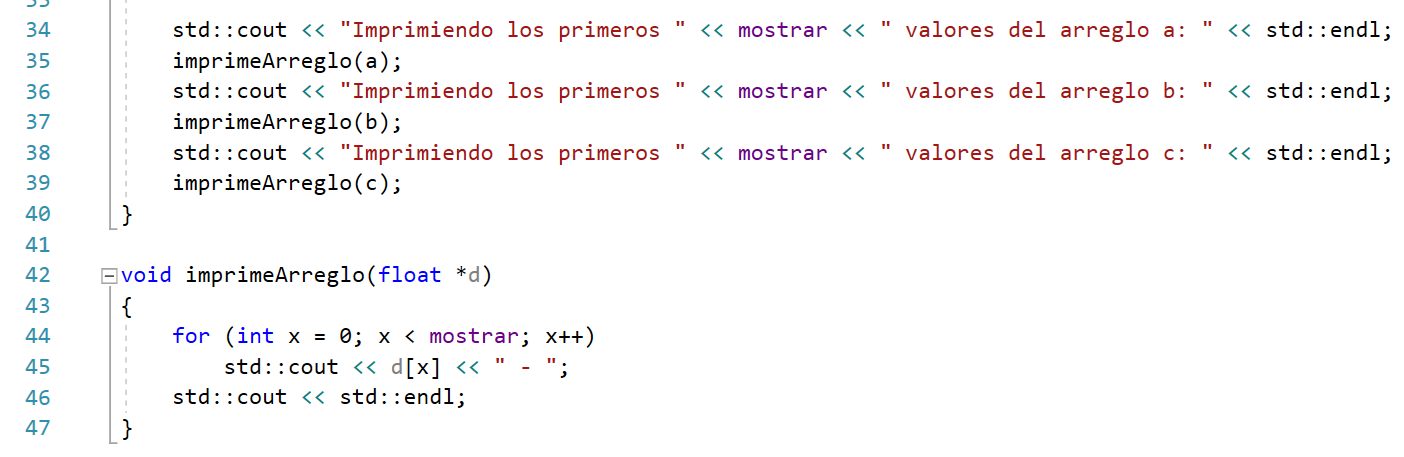
* Declaración de 3 arreglos (a, b, c). Los dos primeros usados para asignar valores aleatorios dentro de un ciclo for. El arreglo “c” almacena el resultado de la suma de los dos primeros.



* Después definimos la instrucción for que se realizara en paralelo usando la librería OpenMP. Este ciclo es donde se realiza la sumatoria en paralelo.



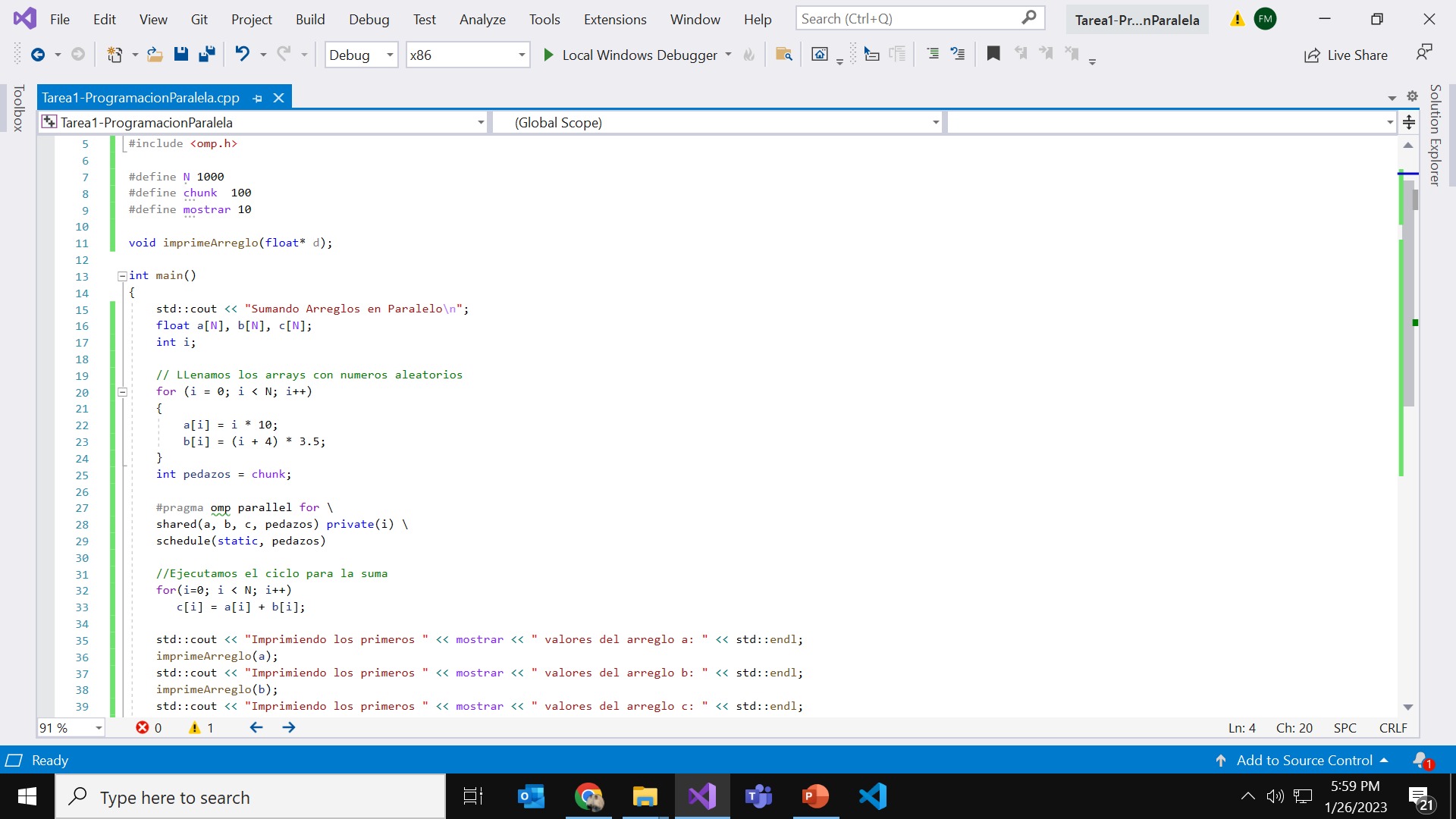
* Luego se imprime la suma en paralelo para comprobar los resultados, imprimiendo tanto los valores de a y b y sus sumas en “c”. Se invoca a la función ImprimeArreglo().

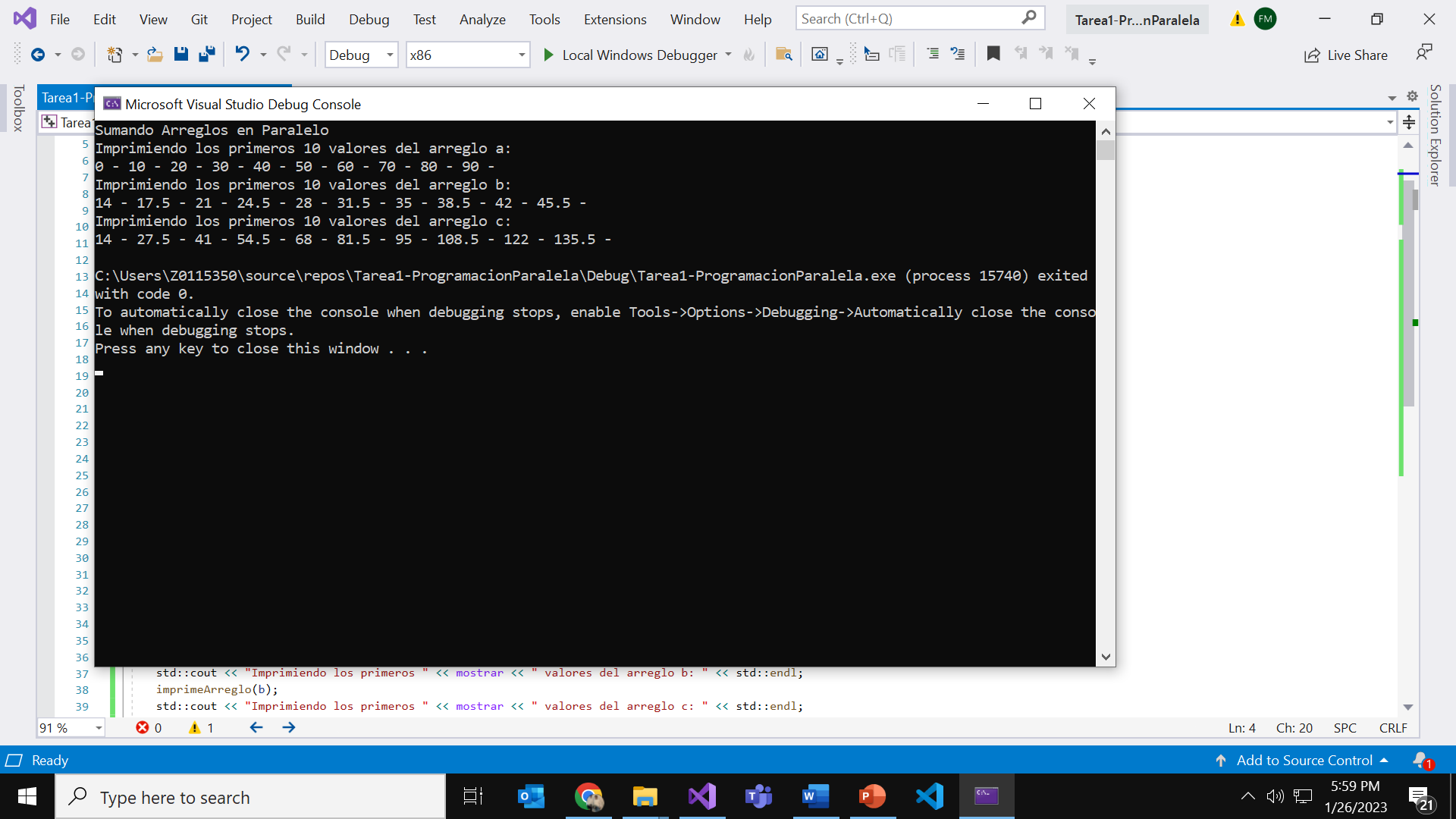


**Ejecución y resultados del código:**

- En la primera ejecución use los valores: N=100, chunk=100, mostrar = 10.

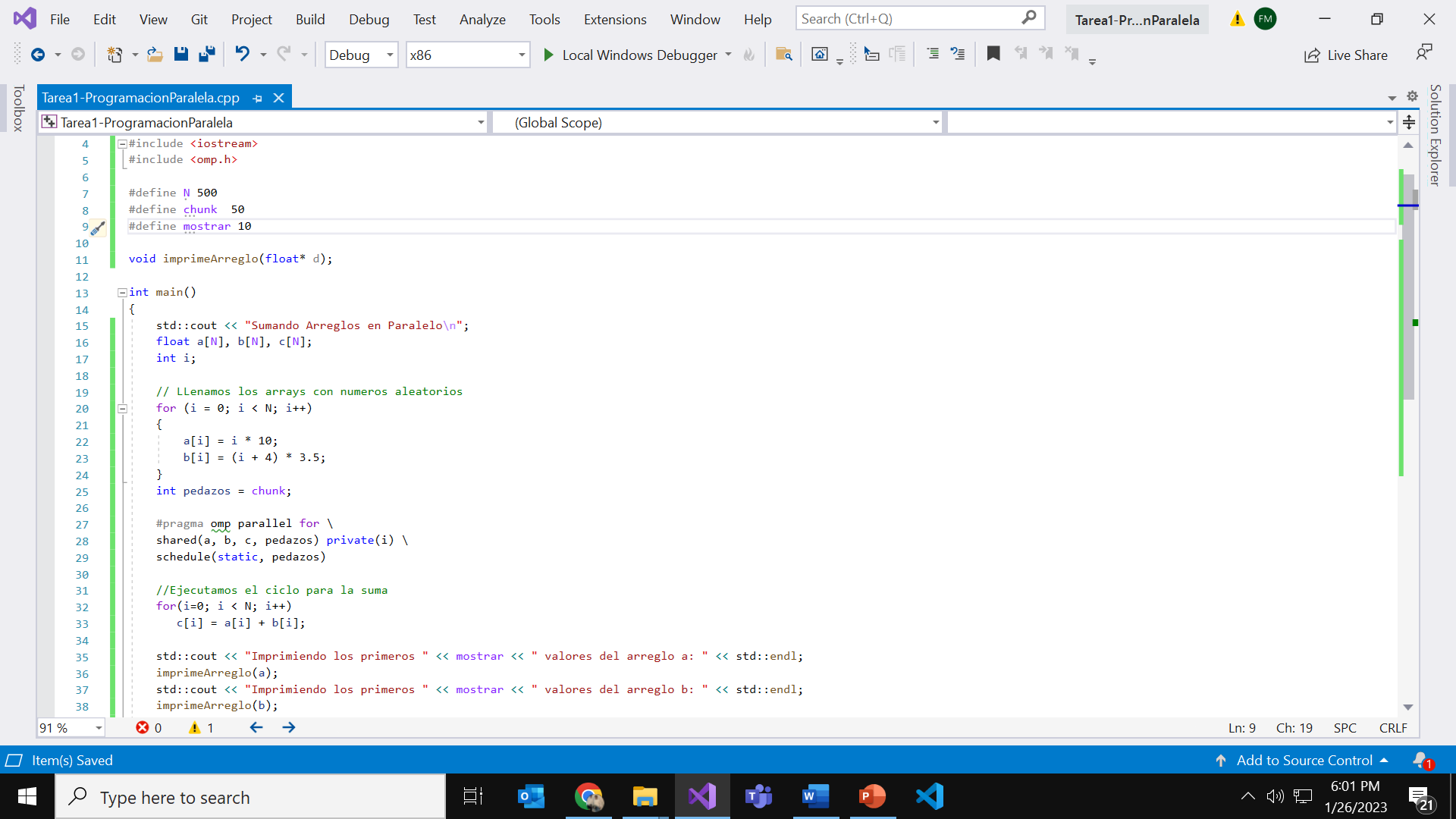
Nótese como las sumas de cada elemento de los arreglos a y b las sumas se muestran correctamente. Ejemplo la suma de a[i]+b[i] es correcta.

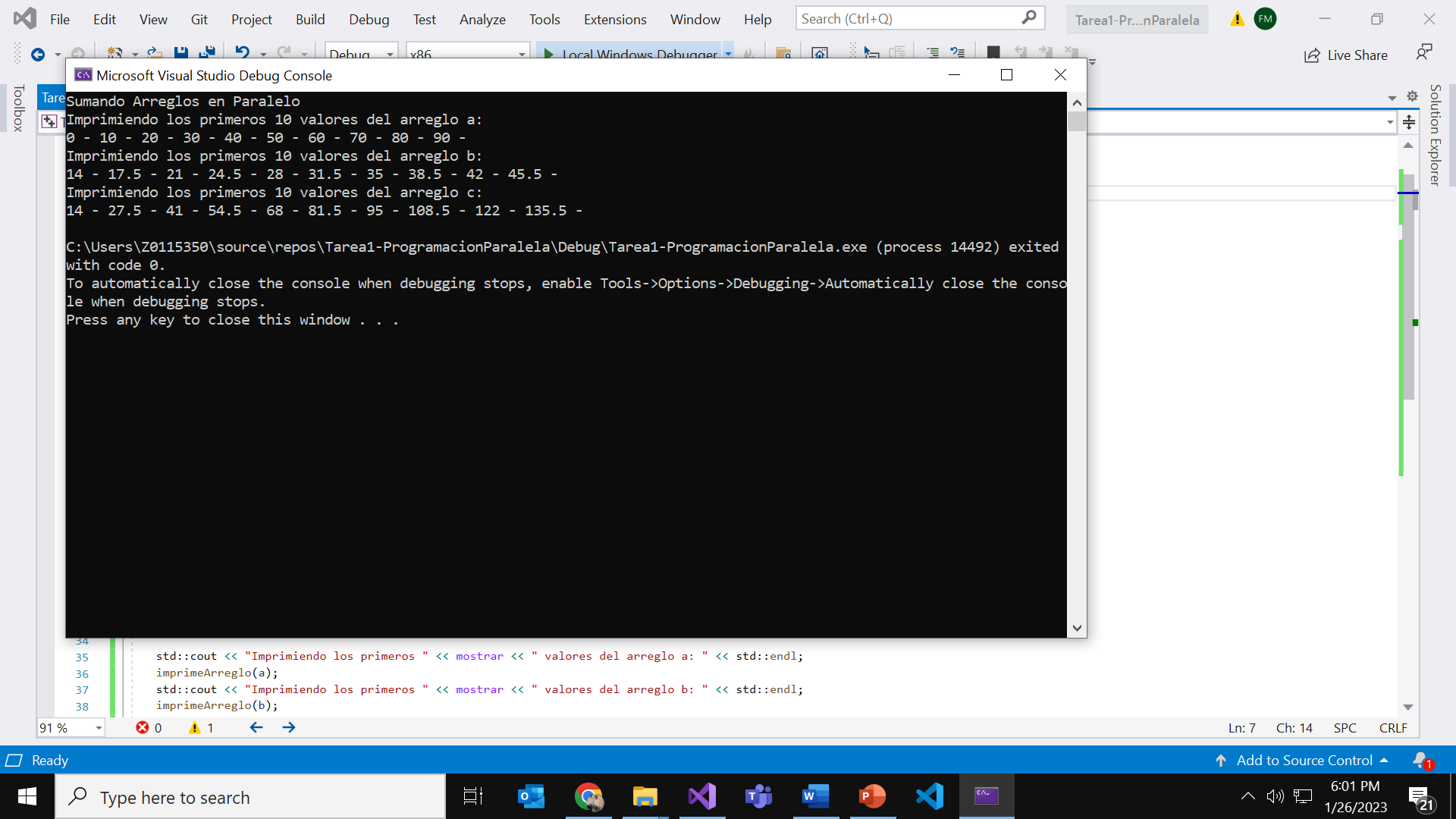


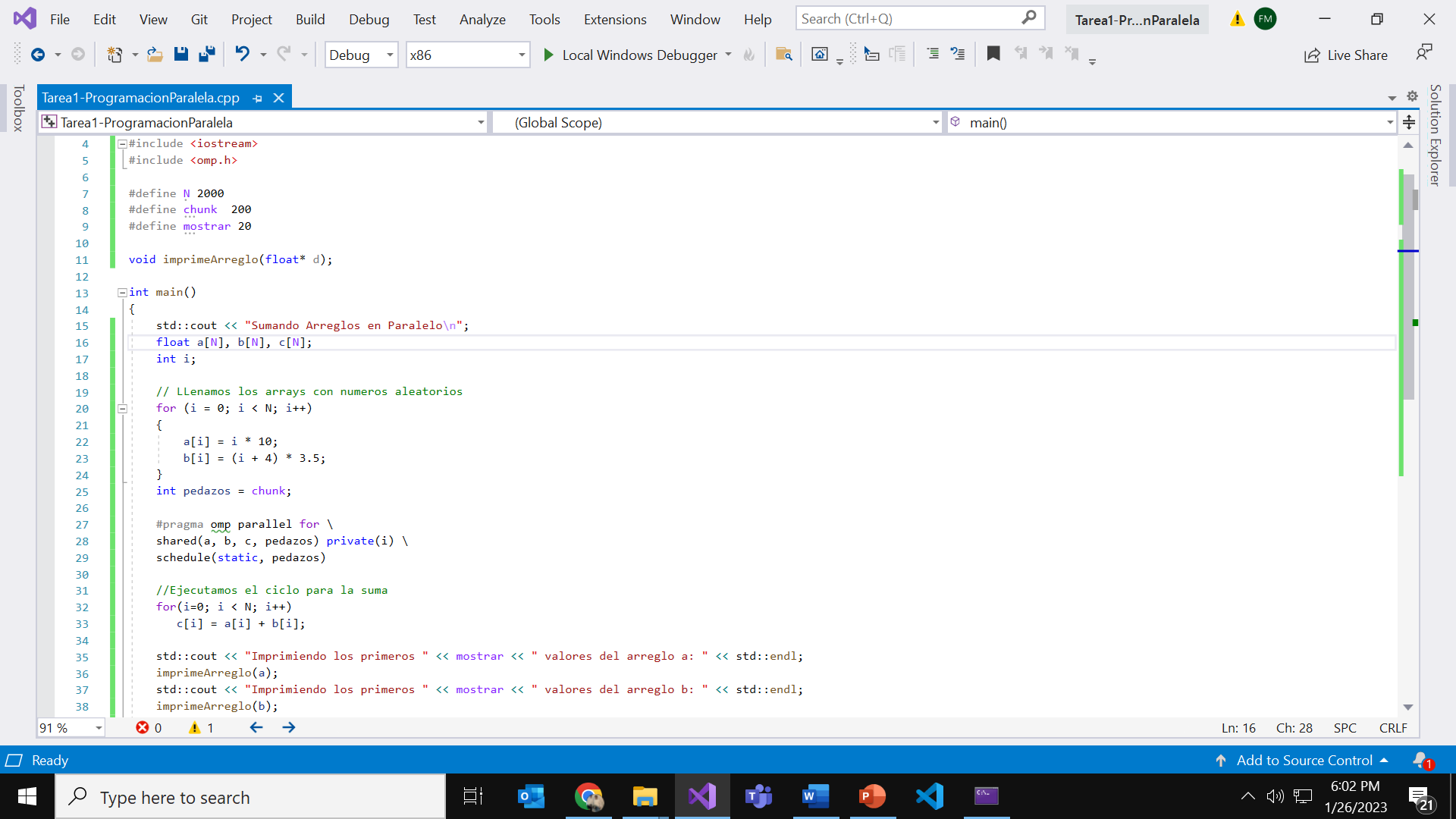


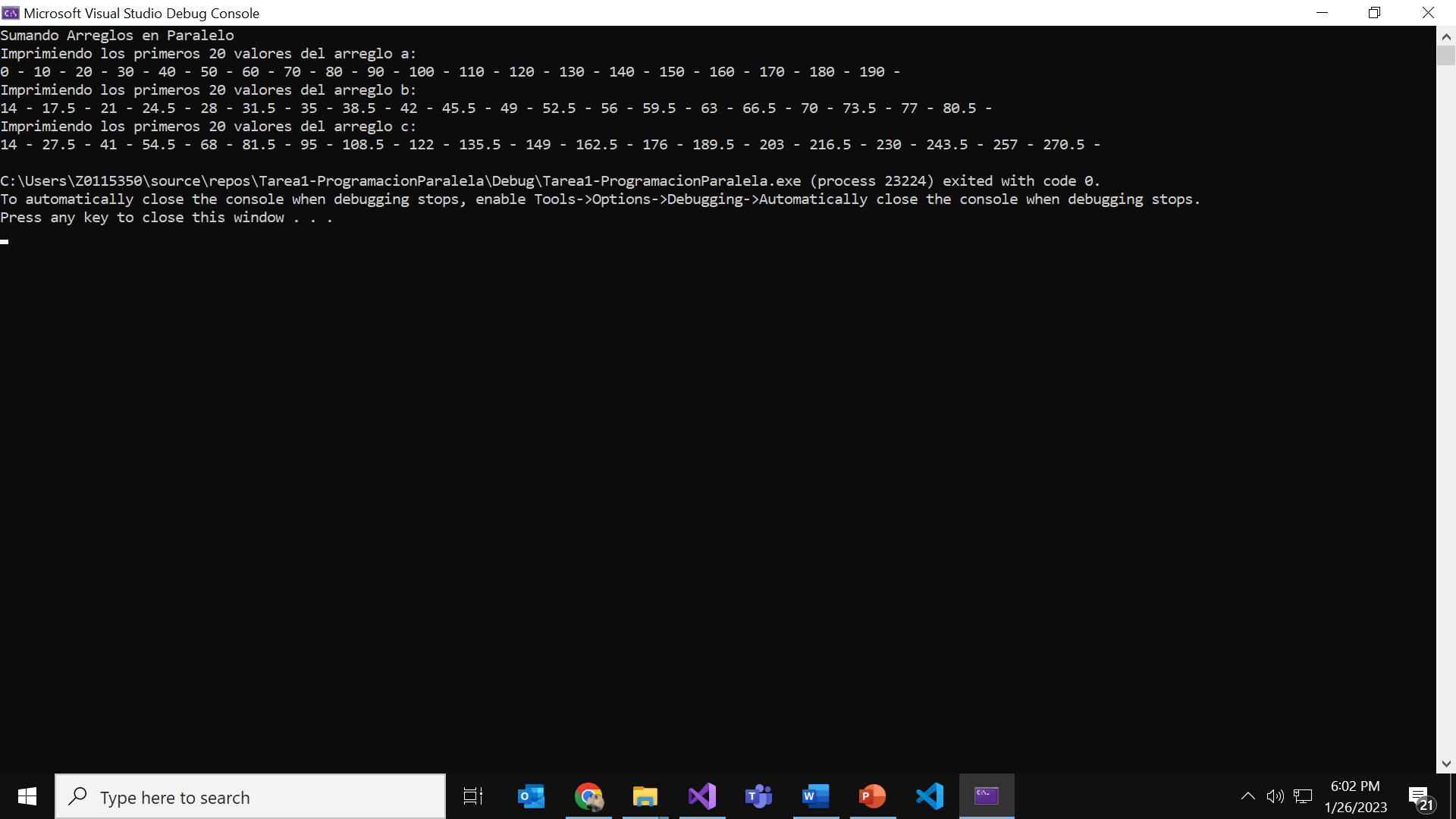
- En la primera ejecución use los valores: N=500, chunk=50, mostrar = 10.

Pude percibir que el tiempo fue un poco mayor debido a los 500 elementos de cada arreglo, pero el incrementar el chunk ayudo a la velocidad de ejecución. Nótese como las sumas de cada elemento de los arreglos a y b las sumas se muestran correctamente. Ejemplo la suma de a[i]+b[i] es correcta.









4. Reflexión de la programación paralela:

La programación paralela ha sido la causa de grandes avances tecnológicos y es una gran técnica se requiere mejorar el rendimiento de un algoritmo. Pero no siempre es la mejor opción ya que no todo se puede ejecutar en paralelo hay tareas que requieren como entrada la salida de otras o tareas que necesitan acceder a los mismos recursos de memoria. También están los costes que se generan el implementar estos algoritmos. Aunque al final del día si hay muchas tareas que pueden ser optimizadas con paralelización.