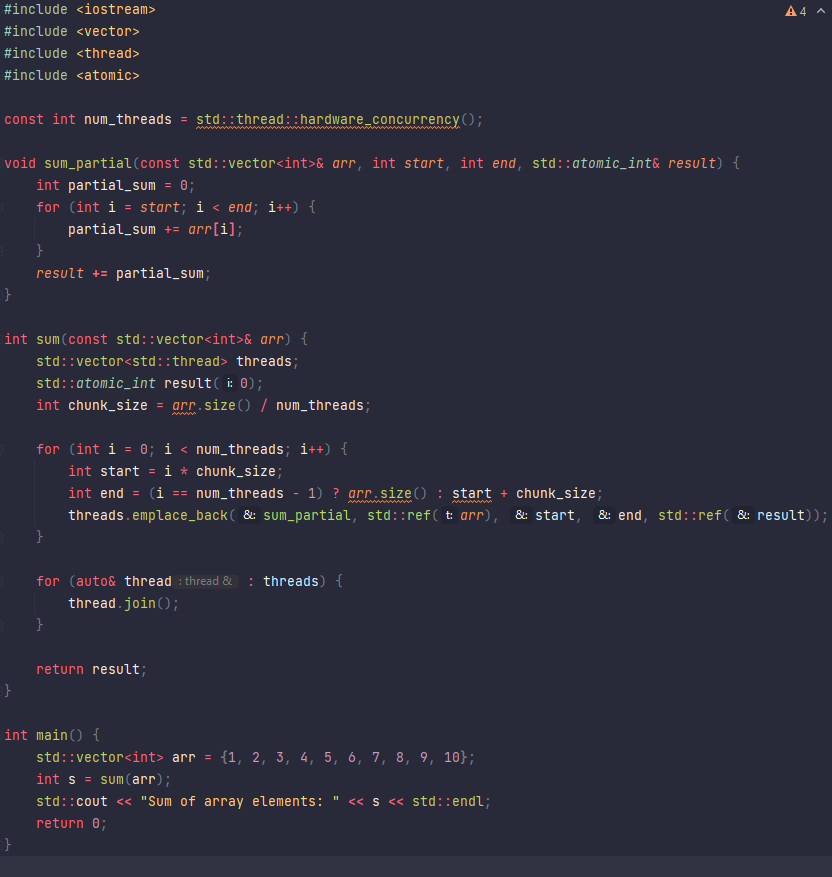
**Laboratorium 7**

**Zad 1**



**Zad 2**



Ten kod implementuje algorytm sumowania elementów w tablicy przy użyciu wielu wątków. Funkcja **sum\_partial** wykonuje częściowe sumowanie elementów tablicy od indeksu początkowego do końcowego indeksu-1. Wynik sumowania przechowywany jest w zmiennej **result**, która jest typu **std::atomic\_int**, co oznacza, że dostęp do niej jest atomowy, czyli jednoczesny odczyt i zapis jest bezpieczny w środowisku wielowątkowym. Funkcja sum przyjmuje jako argument tablicę **arr** i zwraca sumę jej elementów. Najpierw tworzy wektor wątków o rozmiarze równym liczbie dostępnych procesorów, czyli zmiennej **num\_threads**. Następnie dzieli tablicę na kawałki **num\_threads**, z których każdy jest sumowany przez osobny wątek. W pętli for tworzone są wątki, a do każdego z nich przekazywana jest odpowiednio referencja do tablicy **arr**, indeks początkowy, indeks końcowy oraz referencja do zmiennej **result**. Funkcja **emplace\_back** dodaje nowy element na koniec wektora wątków. Po utworzeniu wszystkich wątków program czeka na ich zakończenie korzystając z pętli for i metody join(). Na koniec funkcja sum zwraca sumę elementów tablicy, która jest przechowywana w zmiennej **result**.

Podsumowując, kod ten zrównolegla algorytm sumowania elementów w tablicy poprzez podział tablicy na części i sumowanie każdej części przez osobny wątek. Używa on biblioteki wątków oraz typu atomowego do bezpiecznego dostępu do zmiennej wynikowej w środowisku wielowątkowym.