

Zadanie 1: Trial division

- Napisz funkcję, która przyjmuje liczbę całkowitą i zwraca True jeśli liczba jest liczbą pierwszą, używając metody Trial Division.
- Napisz funkcję, która przyjmuje dwie liczby całkowite start i end i zwraca listę wszystkich liczb pierwszych w tym zakresie, używając Trial Division.
- Napisz funkcję, która przyjmuje dwie liczby całkowite i używa Trial Division, aby znaleźć ich największy wspólny dzielnik (NWD).
- Modyfikacja funkcji trial_division, która oprócz listy dzielników zwraca także liczbę wykonanych kroków (tzn. liczbę sprawdzonych dzielników), aby zbadać wydajność metody.
- Napisz funkcję, która przyjmuje liczbę całkowitą, znajduje jej czynniki pierwsze, używając Trial Division, a następnie sprawdza czy produkt tych czynników jest równy oryginalnej liczbie.

Zadania 2: Sito Eratostenesa

- Napisz funkcję, która używając sita Eratostenesa, znajdzie wszystkie liczby pierwsze mniejsze od zadanej liczby oraz zwróci ich liczbę.
- Napisz funkcję, która przyjmuje liczbę n i wykorzystuje sito Eratostenesa do znalezienia sumy wszystkich liczb pierwszych mniejszych niż n .
- Zaimplementuj wizualizację pracy sita Eratostenesa, gdzie kolejne skreślenia na liście liczb będą wizualnie pokazane, np. za pomocą biblioteki matplotlib w Pythonie.*
- Zmodyfikuj standardową implementację sita Eratostenesa, aby działała efektywniej dla bardzo dużych liczb. Skorzystaj z optymalizacji, takiej jak pomijanie liczb parzystych w tablicy. Porównaj czas wykonania standardowej implementacji sita Eratostenesa z jego zoptymalizowaną wersją dla dużych n (np. 10^6 , 10^7). Użyj modułu time do zmierzenia czasów wykonania.**