### **Опис завдання:**

1. **Автоматизація взаємодії з ШІ:**
   * Ваша програма повинна звертатися до ChatGPT (через OpenAI API або інший аналогічний сервіс), з запитом на створення алгоритму пошуку і виправлення аномальних точок GPS координат мовою **C++**.
   * Отриманий код зберігається у .cpp файл.
2. **Суть задачі для ШІ:**

В наданих файлах з тестовими данними містяться точки з GPS координатами (lat, lon) та час в який координати було зафіксовано (timestamp).

Деякі з цих координат помилкові(аномальні) . Наприклад, стрибок координати на 1000+ метрів за 5 секунд. Це типовий випадок "GPS-сплеску" — неправильного позиціонування, який не збігається ні з напрямком, ні з швидкістю руху.  
Задача яка ставиться перед ШІ - написати алгоритм по виявленню таких координат та їх заміні теоретично коректні (шляхом інтерполяції).

1. **Компіляція та запуск:**
   * Ваш скрипт має автоматично **скомпілювати** отриманий C++ код за допомогою g++(або аналогу).
   * У випадку помилки компіляції — вивести повідомлення та завершити ітерацію.
   * Якщо компіляція успішна — виконати отриману програму, **передавши на вхід пакет данних з GPS координатами**.
2. **Тестування алгоритму:**
   * Запустіть програму для кожного з трьох пакету точок.
   * Після кожного запуску:  
     + перевірте правильність виконання (відсутність аномальних координат, середня та медіанна швидкість руху між точками не сильно відрізняється від максимальної);
     + заміряйте та збережіть **час виконання**;
   * Розрахуйте **середній час виконання алгоритму**.
3. **Зворотний зв’язок для ШІ:**
   * Після кожної ітерації надсилайте ШІ запит з описом:  
     + як працював його попередній алгоритм;
     + середній час виконання;
     + прохання покращити алгоритм.
   * ШІ може змінювати підхід або шукати нові алгоритми через інтернет (за допомогою відповідної інструкції в запиті).
4. **Цикл покращення:**
   * Повторіть цей процес до досягення **15 успішних тестувань створеного коду(тобто код був скомпільований, тестові данні коректно виправлені)**, з кожною ітерацією ШІ повинен намагатися видавати покращену версію алгоритму.
   * Зберігайте результати кожної ітерації (успіх компіляції, успіх пошуку аномальних точок, успіх виправлення точок, час виконання алгоритму, код алгоритму, коментар ШІ що було змінено(опціонально)).
5. **Фінальний звіт:**
   * Після завершення 15 успішних ітерацій:  
     + Виведіть статистику по кожній ітерації;
     + Покажіть, який алгоритм був найефективнішим;
     + Додайте код найкращого алгоритму у фінальний звіт;
     + Додайте короткий висновок, чи вдалося досягти покращення продуктивності за допомогою ШІ.

### **Вимоги до реалізації:**

* Використання Python або С++ як мови для автоматизації (або інша за згодою).
* Генерація та виконання **C++-коду**.
* Компіляція за допомогою g++(чи аналогу).
* Використання OpenAI API або іншого подібного сервісу ШІ.
* Базовий контроль помилок: неправильний код, некоректний помилки компіляції, тощо.

### **Очікуваний результат:**

Посилання на репозиторій (GitHub/GitLab) з:

* + кодом автоматизації,
  + результуючим кодом від ШІ
  + результатами тестів,
  + фінальним звітом,
  + README з інструкціями для запуску;