Цей код реалізує лексичний аналізатор для простої мови програмування, яка підтримує такі елементи, як числа, рядки, коментарі, зарезервовані слова, оператори та розділові знаки. Нижче наведено детальний опис кожної частини коду та його призначення.

### **Основні компоненти**

1. **Підключення бібліотек**:
   * #include <iostream>: для виводу результатів у консоль.
   * #include <fstream>: для можливого використання роботи з файлами (не використовується в даному коді).
   * #include <string>: для роботи з рядками.
   * #include <unordered\_map>: для створення асоціативних масивів (не використовується в даному коді).
   * #include <vector>: для використання динамічних масивів.
   * #include <regex>: для роботи з регулярними виразами.
2. **Перелічення TokenType**:
   * Це перерахування визначає різні типи лексем, які може виявити аналізатор, включаючи: Number, FloatNumber, HexNumber, StringLiteral, CharLiteral, PreprocessorDirective, Comment, ReservedWord, Operator, Punctuation, Identifier та Unknown.
3. **Структура Token**:
   * Ця структура представляє лексему, яка складається з рядка lexeme (саме слово або символ) та його типу type (тип лексеми).
4. **Таблиці зарезервованих слів, операторів і розділових знаків**:
   * reservedWords, operators, punctuation: вектор рядків, що містять зарезервовані слова, оператори та розділові знаки відповідно.

### **Функції**

1. **recognizeTokenType**:
   * Ця функція приймає рядок (lexeme) і визначає його тип, використовуючи регулярні вирази та перевірки на наявність у таблицях зарезервованих слів, операторів і розділових знаків.
   * Наприклад, для чисел використовуються регулярні вирази, що відповідають формату цілих і дійсних чисел, а для рядкових та символьних літералів — відповідні символи (" і ').
2. **tokenTypeToString**:
   * Ця функція перетворює тип лексеми (з TokenType) на рядок для зручного виводу.
3. **lexicalAnalysis**:
   * Головна функція лексичного аналізатора, яка приймає рядок коду (code).
   * Вона проходить по всіх символах у коді, обробляючи їх і формуючи лексеми на основі зазначених правил:
     + Пропускає пробіли, якщо не перебуває в рядкових або символьних літералах.
     + Обробляє рядкові та символьні літерали, коментарі (однорядкові і багаторядкові).
     + Обробляє оператори та розділові знаки.
     + Формує лексеми, поки не зустріне оператор або розділовий знак.
   * Після завершення аналізу виводить всі виявлені лексеми з їхніми типами.

### **Точка входу (main)**

* У точці входу визначається рядок code, який містить код для аналізу (у даному випадку — проста програма на C++, яка оголошує змінні, присвоює їм значення, містить коментарі та має функцію main).
* Функція lexicalAnalysis викликається з цим кодом для виконання аналізу.
* Результати аналізу виводяться на екран.