Problem B: Scanner

請注意不能用 #include<regex> 抓到 BA 0分!!!
請注意不能用 #include<regex> 抓到 BA 0分!!!
請注意不能用 #include<regex> 抓到 BA 0分!!!

Description:

在編譯器的運作過程中，Scanner 是一個核心組件，負責將輸入的文字 (input text) 解析並轉換為 Token。Token 是構成程式碼的最小單位，這些單位必須首先透過 Scanner 辨識出來，才能進行後續的編譯步驟。請參照表一中的定義，實作一個簡單的 Scanner 程式，讓它能夠讀取輸入的文字，並將其轉換為對應的 Token，並輸出這些 Token 以便後續的處理。

Token類型	**Regular Expression**
NUM	**(0|[1-9][0-9]*)**
IDENTIFIER	**[A-Za-z][A-Za-z0-9]***
SYMBOL	**[\+\-*\/\=\(\)\{\}\<\>\;]**
KEYWORD	**(if|while)**

▲表一

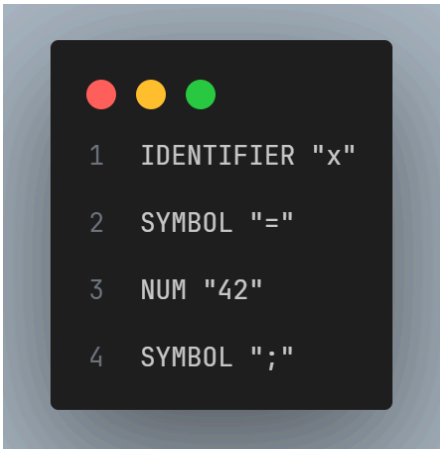
Input Format

1. 輸入是一段程式碼，其中包括**NUM**、**IDENTIFIER**、**SYMBOL**、**KEYWORD**和空白。
2. **NUM**是零或非零開頭的正整數。
3. **IDENTIFIER**是由字母（大小寫皆可）、數字組成的，其中第一個字元一定是字母。
4. **SYMBOL**為單一字元，包含 +、-、*、/、=、(、)、{、}、<、>和 ;。
5. **KEYWORD**包括 if 和 while。
6. 當輸入文本中包含任意數量的空白字元時（如空格、換行符號等等），這些空白字元會被忽略。像是空格（ASCII 32）、換行符號 '\n'（ASCII 10）和 '\r'（ASCII 13）等都會被跳過，不會被 Scanner 解析。
7. 部分輸入的程式碼將會有不符合 Regular Expression 的情況發生。

Output Format:

請在切割 input 後輸出其 Token 類型。

1. Output 應該以純文字的形式呈現，每個Token佔據一行。
2. 每一行的格式為：類型 "內容"，並以一個空白做為區隔。「類型」是該Token的類型，「內容」是該Token的具體內容，例如：**SYMBOL "="**。請參考下圖：



3. 若 input 不符合Regular Expression，則該行輸出只需印出：**Invalid**。請參考下圖：

Sample input 1 x = 42;	**Sample output 1** IDENTIFIER "x" SYMBOL "=" NUM "42" SYMBOL ";"
Sample input 2 x = ^ 42 $;	**Sample output 2** IDENTIFIER "x" SYMBOL "=" Invalid NUM "42" Invalid SYMBOL ";"

Sample input 3
while (x <= 999){
  x = x + 1;
}
	Sample output 3
	KEYWORD "while"
	SYMBOL "("
	IDENTIFIER "x"
	SYMBOL "<"
	SYMBOL "="
	NUM "999"
	SYMBOL ")"
	SYMBOL "{"
	IDENTIFIER "x"
	SYMBOL "="
	IDENTIFIER "x"
	SYMBOL "+"
	NUM "1"
	SYMBOL ";"
	SYMBOL "}"

Sample input 4
if (number1 < 100){
  number1 = number2 + 1;
}
	Sample output 4
	KEYWORD "if"
	SYMBOL "("
	IDENTIFIER "number1"
	SYMBOL "<"
	NUM "100"
	SYMBOL ")"
	SYMBOL "{"
	IDENTIFIER "number1"
	SYMBOL "="
	IDENTIFIER "number2"
	SYMBOL "+"
	NUM "1"
	SYMBOL ";"
	SYMBOL "}"