Assignment 01

Problem Statement 01:

Write a **Lexical Analyzer** using **Flex and Bison** that reads a given text (sentence or code snippet) and performs the following tasks:

1. **Detect all identifiers**
   - A valid identifier starts with a letter or underscore (_) followed by letters, digits, or underscores.
   - Example: x, my_var1, _temp.
2. **Detect all constants**
   - A constant is a sequence of digits (0-9) without any letters.
   - Example: 123, 7890.
3. **Detect invalid tokens**
   - An invalid token is a word that **starts with a number but contains letters afterward** (e.g., 12abc, 45x). Such tokens are neither valid identifiers nor constants.
4. **Ignore all special symbols and punctuation** (like =, ;, .) and whitespace.

Example Input:

```
x = 123; 12abc y1 45 9z;
```

Expected Output:

```
Identifier: x

Constant: 123

Invalid token: 12abc

Identifier: y1

Constant: 45

Invalid token: 9z
```

Problem Statement 02:

Write a Lexical Analyzer using Flex and Bison that reads multiple sentences from the user input and performs the following tasks:

1. Detects each sentence entered by the user.
2. Considers a **sentence** as any sequence of characters ending with a **period (.)**.
3. Prints each detected sentence on a new line, along with its **line number** (e.g., *Line 1*, *Line 2*, etc.).
4. At the end of the input, displays the **total number of sentences (lines)** detected.
5. Ignores extra whitespace (spaces, tabs, or newlines).

Example Input:

```
I love programming. Flex and Bison are useful tools.

This is the third sentence.
```

Expected Output:

```
Line 1: I love programming

Line 2: Flex and Bison are useful tools

Line 3: This is the third sentence



Total number of lines detected: 3
```