# Recursive!!!

## Description

Please write a program to read the source code as input. Your program should have this two functionality:

(1) To identify token which defined in Table 1

(2) To check input (source code) have follow the grammar rule as belong or not.

***Hint:*** *You should implement two phase of compiler, first is scanner and second is parser. (if you guys don't know why, see you in 2019……)*

**Important - 必須使用 Recursive-Decent-Parsing(ch02 ppt page.20 begin) 的模式來撰寫程式，否則將會不予計分！**

▼ Table 1 Token Definition

| Terminal | Regular Expression |
|---|---|
| **id** | [A-Za-z]+ |
| **assign** | = |
| **inum** | [-]?[0-9]+ |

▼ Grammar rule

```
1.  Procs → Val $

2.  Val   → id assign Val
          | inum
```

## Input Format

The input is a string sequence; it must have a space character in any two continuous string.

The length of input sequence is less than one thousand character.

The character of each string is less than ten.

## Output Format

Your task is to check input sequence is valid or not. If input sequence can't pass Scanner phase, it should not go into Parser phase.

1. If the source code can pass both scanner and parser phase, please print **TA is handsome.** (and a new line character)
2. If the input doesn't match token definition, please print **Token mismatch.** (and a new line character)
3. If the input doesn't match grammar rule, please print **Grammar mismatch.** (and a new line character)

| | |
|---|---|
| **Sample Input**<br>a = Bb = -70<br>**Sample Output**<br>TA is handsome.↵ | **Sample Input**<br>a = 70 = Bb<br>**Sample Output**<br>Grammar mismatch.↵ |
| **Sample Input**<br>a1 = 70<br>**Sample Output**<br>Token mismatch.↵ | **Sample Input**<br>a = 70 = c1<br>**Sample Output**<br>Token mismatch.↵ |

這一題每個字之間都有空白，不必用 getline 讀一整行再去切割，可以參考以下方法

| | |
|---|---|
| C<br>char str[???];<br>while(~scanf(“%s”, str)) {<br>    //do something<br>} | C++<br>while(!cin.eof()) {<br>    string str;<br>    cin>>str;<br>    //do something<br>} |