20230413 計算機概論9 Assignment

**題目：**

Token是一個字元串，是構成code的最小單位，從輸入字元流中擷取token的過程叫作Tokenization。舉例來說，如果輸入為**sum=3+2;**這一字串，那將其Tokenization後可以得到下表內容：

|  |  |
| --- | --- |
| Token | Token類型 |
| sum | Identifier |
| = | Special Symbol |
| 3 | Number |
| + | Special Symbol |
| 2 | Number |
| ; | Special Symbol |

這次的作業希望你基於Parctice 10，加上對輸入字元流做Tokenization的功能，使用**Java的class**寫出一個能夠**將一串簡單C語言程式碼做Tokenization，並根據以下三個case做分類**的程式。

Token的種類：

Case1：Identifier

底線或英文字母開頭，之後跟著英文字母或底線或數字

Ex：int, \_abc, k9999, a, cc, kfn\_0, a2b, return, ...

Case2：Number

一般的數字，測資不會有包含小數點的浮點數出現。

Ex：1, 23, 284, 295, 7729...

Case3：Special Symbol (以下所列出來的33種特殊符號)

+ - \* / > < = % & | ^ "

' . , ( ) [ ] { } ! : ;

++ -- += <= >= == != && ||

你的Lexical Analyzer會根據以上三種case將讀入的input做分類，接著將會根據讀入的command執行以下動作：

0. 結束程式

1. 印出總共多少個token

2. 印出所有的token

3. 印出三種case各有多少token

Input的方式採用螢幕輸入，會輸入一段長度不定的程式碼，以END\_OF\_FILE 這個字做結尾。給予的input裡可能混雜著長度不定的white space(空白,Tab,換行)，要無視那些white space並且切出正確的token。接著會輸入數個0~3的數字，來測試互動式功能選項。

**題目要求：**

Class **Token**中需包含有以下元件(程式中必須使用此class 否則0分)

|  |  |  |  |
| --- | --- | --- | --- |
| 名稱 | 權限 | 型態 | 說明 |
| token | private | string | 字元串 |
| type | private | string | Token種類 |
| get\_token | public | Void function() | 從input中抓取一個token並且存入this.token裡面。 |
| get\_token\_type | public | Void function() | 判斷你的this.token是哪種種類的token。 |

The actual input and output for running your program is something that looks like the following：

**例一：**

# input自此始，但不包括此行

void main() {

int a=5;

int b=10;

int sum=a+b;

}

END\_OF\_FILE

1

2

3

0

# input至此止，但不包括此comment，每行後面都有line-enter

# output自此始，但不包括此行

Total number of tokens: 23

[void]

[main]

[(]

[)]

[{]

[int]

[a]

[=]

[5]

[;]

[int]

[b]

[=]

[10]

[;]

[int]

[sum]

[=]

[a]

[+]

[b]

[;]

[}]

Identifier: 10, Number: 2, Special Symbol: 11

# output至此止，但不包括此comment



在terminal執行程式的結果圖

**例二：**

# input自此始，但不包括此行

aaaa\_2jsji = 45;

\_\_www== 89;

+ = ;

a [30] = 2\_ab

END\_OF\_FILE

3

1

2

2875

0

3

# input至此止，但不包括此comment

# output自此始，但不包括此行

Identifier: 4, Number: 4, Special Symbol: 10

Total number of tokens: 18

[aaaa\_2jsji]

[=]

[45]

[;]

[\_\_www]

[==]

[89]

[;]

[+]

[=]

[;]

[a]

[[]

[30]

[]]

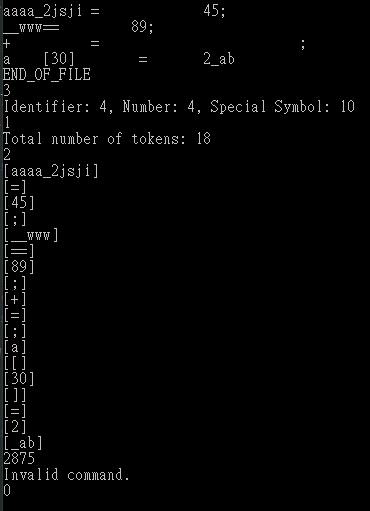
[=]

[2]

[\_ab]

Invalid command.

# output至此止，但不包括此comment



在輸入0之後，你的程式就要停止運作，不該有其他的動作

**補充說明：**

1. 測資不會有\n、\t等字符出現。

2. 不會有+++或是---之類的情況發生。

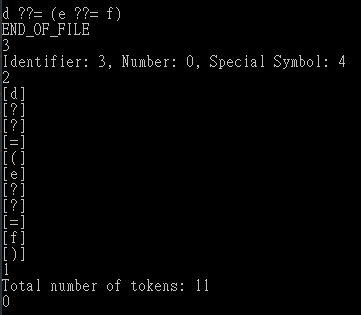
3. 測資不會有”string”出現（ex: “Hello World”, “This is 1 string”）。

4. 數字不包含小數點。

5. 指令的部分，輸入0之後程式應該立即停止運作，不再讀入任何東西。

6. 一個token不會超過30個字元。

7. 若有unknown token(ex: ?, ~, @....)出現，你的程式會將他們當作單一token讀入，但不做任何處理。舉例來說，如果輸入為sum?sum = 20;，你的程式應該會切割出[sum], [?], [sum], [=], [20], [;]。但unknown token不會被算進case 1, case2, 或case3之中，詳細輸出範例請見下圖。



處理unknown token之範例輸出

**評分標準：**

For each input (test data) file, your program must produce an output file with a content that is EXACTLY THE SAME AS the expected output. This is the only way your program can "pass" any particular test.

In general, we will use 5 test data to test your project. The total score of this project is 100 points, and each test data accounts for 20%.

**解題指引：**

\* 該如何分割token ?

從下一個尚未讀進來的或已讀進來但尚未處理的字元(char)開始，就是在概念 上尚未處理到它的那個字元(char)。

1. 跳過所有white-space(空白,Tab,換行)、直到碰到「非white-space」
2. 你已經得到下一個token的第一個char，可以從這個char來判斷接下來我們會讀入的這個token可能為哪一case
3. 從那個非white-space字元開始讀入、且盡可能的往後讀，直到碰到white-space或前面所提到的特殊字元為止