ParserTest

Description

Please write a program to read a program source from stdin following the token definition and grammar rule at right.
If **Yes**, print each **token's type** and **the string of token** seperated by **a whitespace** “ ” and end with a **newline**.
If **No**, print only **“invalid input”** with **a newline**. (don't output any token!)

Your program has to check the source whether it follows the token and grammar rules or not.

※請使用 Recursive-Decent-Parsing (ch02 ppt page 20 begin) 的模式來撰寫程式，否則將不予計分。

※測試檔案的換行皆為 \n

Terminal	Regular Expression
ID	[A-Za-z_][A-Za-z0-9_]*
STRLIT	“[^”]*”
LBR	\(
RBR	\)
DOT	\.
SEMICOLON	;

Productions

1	program	→	stmts
2	stmts	→	stmt stmts
3	stmts	→	λ
4	stmts	→	exp SEMICOLON
5	exp	→	primary
6	exp	→	STRLIT
7	exp	→	λ
8	primary	→	ID primary_tail
9	primary_tail	→	DOT ID primary_tail
10	primary_tai	→	LBR exp RBR primary_tail
11	primary_tai	→	λ

Sample Input "test_string"; Test_ID; **Sample Output** STRLIT " test_string " SEMICOLON ; ID Test_ID SEMICOLON ;	**Sample Input** illiga!id; **Sample Output** invalid input
Sample Input Str. length(); **Sample Output** ID Str DOT . ID length LBR (RBR) SEMICOLON ;	**Sample Input** printf(“HelloWorld”); **Sample Output** ID printf LBR (STRLIT " HelloWorld " RBR) SEMICOLON ;

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
char str[???];
while(~scanf(“%s”, str)) {
        //do something
}
```

```
C++
while(!cin.eof()) {
        string str;
        cin>>str;
        //do something
}
```