

1. Write a Lex program to parse a CSV(comma separated values) file. The CSV file will also contain comments(beginning with # or ; or !). A line beginning with #;! is counted as a comment only if one of these characters occurs at the beginning of the line. Detailed explanation of input and outputs follows.

Input:

```
# this is a comment.
```

```
field1,field2 , field3,field4 , field5 ,field6
,field2,, ,field5,
! Another Comment
!notcomment
```

Output:

```
3 field1
3 field2
3 field3
3 field4
3 field5
3 field6
4
4 field2
4
4
4 field5
4
6 !notcomment
```

Output is of the format <line\_number><SPACE><CSV Field>, each CSV field is printed on a separate line.

Note that if a line is empty or a comment you should not include it in the output.

Also, while printing the fields, ignore any whitespace(as indicated in example)

You may only use lex tool in this assignment

2. Consider a simple matrix manipulation language KGPLAB. KGPLAB has 4 types of statements.

- Matrix definition statement - DEF <matrix\_name> [<matrix\_x>,<matrix\_y>];  
eg - DEF A [2,3]; // defines a 2X3 Matrix A.  
You can also define multiple matrices by adding a comma after every definition.  
eg - DEF A[2,3] , B[3,4]; //defines a 2X3 matrix A and 3X4 Matrix B
- Matrix Initialization statement - INIT <matrix\_name> [<matrix contents in row-major order>]  
eg - INIT A [1,2,3 : 4,5,6] // Since A is a 2X3 matrix, it will have 2 rows and 3 columns, the contents are specified in row major order with a comma between every entry and colon(:)

after the end of each row.

- Matrix operations - The supported operations are ADD, MMUL and CMUL.
  - ADD <output>,<input\_1>,<input\_2>; - Adds the input matrices and puts the result in <output>  
eg - ADD C,B,A; // C = A+B
  - MMUL <output>,<input\_1>,<input\_2>; - Multiplies 2 matrices and puts the result in <output>  
eg - MMUL C,B,A; // C = A X B
  - CMUL <output>,<matrix>,<constant>; - Multiplies a matrix with a constant and puts the result in <output>  
eg - CMUL B,A,5; // B = A X 5
- Print statement - Prints the contents of the matrix in row-major order.  
eg - PRINT A; // Prints contents of A  
Note that for this statement, multiple arguments are NOT allowed i.e.  
PRINT A,B; // NOT valid
- Comments - Any string after the “//” symbol till the end-of-line.All comments are single-line only  
eg - //This is a comment  
// this is start of a comment  
but this is not a comment

eg - // This is a full line comment

ADD C,B,A; // This is a half-line comment

Your task for this assignment is to tokenize a given KGPLAB program. You should use the following tokens-

<INT> - An integer (used in CMUL or init statement)

<ID> - An identifier (eg - Matrix name)

<ADD> - The add statement

<MMUL> - The MMUL statement

<CMUL> - The CMUL statement

<DEF> - The DEF statement

<INIT> - The INIT statement

<PRINT> - Print statement

<COMMA> - for a comma(,)

<ENDS> - For a end statement semicolon(;

<ENDR> - For an end row colon(:)

<SQBKTO> - Opening Square-Bracket ( [ )

<SQBKTC> - Closing Square-Bracket ( ] )

Note that KGPLAB is **whitespace agnostic**, the tokenization of any program will not be affected by adding or removing whitespace including newline(except in the case of statements and comments).

eg - ADD 2 ,3; and ADD 2,3; are the same but ADD 2,3; and ADD2 , 3; are not the same, in

the second case you should throw an error.

eg - ADD 2,  
3 ; // This is a valid statement

Also **strip all comments** from your output.

Print every statement in a separate line and strip all whitespace from your output.

Example Input -

```
DEF A [2,2];  
INIT A [1,2 : 3,4];  
// Some Comment
```

Example Output -

```
<DEF><ID><SQBKTO><INT><COMMA><INT><SQBKTC><ENDS>  
<INIT><ID><SQBKTO><INT><COMMA><INT><ENDR><INT><COMMA><INT><SQBKTC><E  
NDS>
```

### **Bonus - Input Validation :**

- Validate that the matrix when initialised has the same dimension that was defined.  
eg - DEF A [2,2];  
    INIT A [1,2:3,4]; // Correct  
    INIT A [1,2,3 : 4]; // Incorrect
- Validate the input parameter to matrix operation statements i.e. Validate that ADD has all parameters of the same dimension and that MMUL has correct dimension of output matrix.