Project Requirements:

1. Develop a system where you create your own API consumption service which takes off all the load of the processing that would otherwise be required.
2. Input: JSON data as standard input (as per a specified format)
3. Once every data has been input, you print the data to the standard output

Project Description:

1. Enter no. of test cases : n
2. For every test case input one single line consisting of two numbers: nT and nD
    a. nT - no. of tables you are expected to find in the JSON data
    b. nD - no. of JSON data objects that you will expect to be input by the user
3. next line will contain nT - no. of space-separated names of the objects
    a. The object may occasionally be accompanied by a set of parenthesis '()' where the order of the data may be specified.
    b. Order the data only in that specific order (asc or desc)
4. The next lines will be the actual JSON data
    a. Program will expect nD no. of separate JSON data.
5. Maintain an auto-generated id for every object item that you generate

Data Types:

1. Tests - Linked List<> of tests
2. Test - ArrayList<> of tables
3. Table - ArrayList<> of name, order, columnList, type(JSONArray/JSONObject)
4. ColumnList - ArrayList<> of Columns
5. Column - ArrayList<> of name and ColumnData
6. ColumnData -
    a. HashMap<> for JSONObject
    b. ArrayList<> for JSONArray

Coding Description:

1. Read n
2. Loop through n test cases

For each Test case

1. Read nT and nD
2. Read table names
3. Parse table names, and get names and orders
4. Create nT tables
5. Read line by line using while(true)
6. Ignore - if {
7. Break loop - if }
8. Keep track of initial table - table 0
9. Split line using : into key-value pair

a. If value is String, insert in table 0
b. If value is { - inner JSONObject
c. If value is [- inner JSONArray

If inner JSONObject

1. Find table with key
2. In table 0, add new column with this table name /key
3. Read line by line using while(true)
4. Break loop - if },
5. Split line using :
   a. Put value in correct key column in this table
   b. In table 0 column (key) : for current object, put index of current column row

If inner JSONArray

1. Find table with key
2. Add extra column with table 0 name
3. Add new column with current table name
4. Read line by line using while(true)
5. Break look - if]
6. For every input
   a. Add object index in column 0
   b. Add value to column 1

Handle missing column

1. Keep track of prev column name, if first column prev column will be null
2. If column not present:
   a. If prev column null then just add the column
   b. If prev column not null, then add new column after prev column

Print

1. Print test no.
2. Print table name
3. Print column names
4. Loop through table depending on correct order
   a. If JSONObject call printJSONObject
      i. Use stringbuilder
      ii. Loop through per object
   b. If JSONARray call printJSONArray
      i. User StringBuilder
      ii. Loop through per column item
   c. Use stringbuilder:
      i. Print column names
5. Print white line
6. Repeat till all tables done