Problem 8 – What's My Type?

Big Global Distribution ships products coming from and going to a large number of different import/export client companies. Soon they will be performing a data import to a new system. Fortunately, the negotiations between Big Global Distribution and their clients resulted in everyone agreeing on using simple flat files for the data. Unfortunately, no one thought to negotiate the formatting of the files. And now the data is already coming in.

Aloysius, your manager, has tasked your team with composing an application that will analyze the inventory data submitted by the top clients and produce a description of what data types each record contains. Also provide a description of the data types for the entire dataset.

Key Points

- Line feeds and carriage returns are line and record delimiters. They will not be a part of data within the records.
- The pattern of three *at signs* "@@@" (without quotes) on a single line delimits each data set. They will also be present at the end of the last data set.
- The first line of information is the header. The header contains two characters: The *field delimiter* character followed by the *text qualifier* character. Each line following the header is considered a record of data within the entire data set.
- Each record will contain the same number of fields delimited by the *field delimiter* character. The *field delimiter* is the character used in records to separate fields within the record. The *text qualifier* is the character used to surround a field to indicate explicitly that the field is of TEXT datatype.
- The *field delimiter* and the *text qualifier* can be any "printable" ASCII character with decimal values 33 through 126 as well as space (32) and horizontal tab (9) characters.
- Only two data types are defined: NUMERIC and TEXT. A field is considered NUMERIC if it is composed of only numeric characters 0 through 9. Decimals and a leading minus sign are considered numeric characters. A field is considered TEXT if it contains characters that are not numeric and/or if the field is surrounded using *text qualifier* characters.
- Note that not all TEXT fields require *text qualifiers*. The only time a TEXT field requires *text qualifiers* is when the data of the field itself contains a *field delimiter* character or a *text qualifier character*. If a field begins with a *text qualifier* it will always end with a *text qualifier*.

- Fields with embedded characters that are the same as the *text qualifier* character must be surrounded by *text qualifier* characters and the embedded character is represented as a pair of the character.
  The following is an example of a TEXT field with * as the *text qualifier* and embedded characters:
  1000,*TestData*,123,*This field contains **Embedded** characters*,9999

- No input data will be malformed.

Input

Input will be provided through stdin. The input will consist of multiple data sets and will be formatted as described above in the 'Key Points' section. An empty line will signal the end of input.

Output

The output will consist of a row representing each record from the data set indicating what the data type of each field discovered for that record. Each row will be in the following format:

```
Rec #: TEXT,NUMERIC,TEXT
```

- Single space between "Rec" and the record number (starting with 1).
- Single space between the colon and the first field data type.
- New line after the last field data type.

At the end of a data set, output the assumed data type for the field for the entire data set. If a field is the same data type across all records, then the field type for the entire dataset is that type. If a field is of type TEXT for any record, then the field type for the entire dataset is TEXT. Output this line in the following format:

```
Dataset #: TEXT,NUMERIC,TEXT
```

- Single space between "Dataset" and the record number (starting with 1).
- Single space between the colon and the first field data type.
- New line after the last field data type.

Note specifically the last data field of each record and the corresponding data set result in the examples below. Green indicates a TEXT field and blue indicates a NUMERIC field.

Sample Input

```
,"
1000,123,SUPER FUN BALL,SUPER CO.,344 SOMETHING STREET,"SPRINGFIELD, MI 12345",25.23,1112,32542
1001,3,""ELEFUNKY"" PLUSH ANIMAL","DUMBO, LLC",1521 STREET RD,"SPRINGFIELD, MI 12345",22.5,1112,"32542"
@@@
|'
1000|123|SUPER FUN BALL|SUPER CO.|344 SOMETHING STREET|'SPRINGFIELD, MI 12345'|25.23|1112|32542
1001|3|'''ELEFUNKY'' PLUSH ANIMAL'|'DUMBO, LLC'|1521 STREET RD|"SPRINGFIELD, MI 12345"|22.5|1112|32542
@@@
,*
1000,123,SUPER FUN BALL,SUPER CO.,344 SOMETHING STREET,*SPRINGFIELD, MI 12345*,25.23,1112,32542
1001,3,*"ELEFUNKY" PLUSH ANIMAL*,*Q**Bert, LLC*,1521 STREET RD,*SPRINGFIELD, MI 12345*,22.5,1112,32542
@@@
```

Output for the Sample Input

```
Rec 1: NUMERIC,NUMERIC,TEXT,TEXT,TEXT,TEXT,NUMERIC,NUMERIC,TEXT
Rec 2: NUMERIC,NUMERIC,TEXT,TEXT,TEXT,TEXT,NUMERIC,NUMERIC,TEXT
Dataset 1: NUMERIC,NUMERIC,TEXT,TEXT,TEXT,TEXT,NUMERIC,NUMERIC,TEXT
Rec 1: NUMERIC,NUMERIC,TEXT,TEXT,TEXT,TEXT,TEXT,NUMERIC,NUMERIC,NUMERIC
Rec 2: NUMERIC,NUMERIC,TEXT,TEXT,TEXT,TEXT,NUMERIC,NUMERIC,NUMERIC
DataSet 2: NUMERIC,NUMERIC,TEXT,TEXT,TEXT,TEXT,TEXT,NUMERIC,NUMERIC,NUMERIC
Rec 1: NUMERIC,NUMERIC,TEXT,TEXT,TEXT,TEXT,TEXT,TEXT,NUMERIC,NUMERIC,TEXT
Rec 2: NUMERIC,NUMERIC,TEXT,TEXT,TEXT,TEXT,NUMERIC,NUMERIC,NUMERIC
DataSet 3: NUMERIC,NUMERIC,TEXT,TEXT,TEXT,TEXT,TEXT,NUMERIC,NUMERIC,TEXT
```