UCF "Practice" Local Contest — Aug 25, 2012

Lifeform Detector

Filename: `lifeform`

Government scientists at Area 51 are developing a new program to detect alien lifeforms. In particular, they are interested in finding evidence of additional visits by aliens in the 1950's. From their first visit, the scientists have the make-up of the aliens' DNA (which is radically different from human DNA). The complete grammar for the DNA is as follows (lower-case letters represent terminal symbols and ε is the empty string):

`<S> → a<T>b<S> | c<S> | ε`

`<T> → a<T>b<S> | c<S>`

The Problem:

Given possible alien DNA patterns, determine if they match the alien DNA description given by the grammar.

The Input:

The first line of the input will consist of a positive integer *n*, representing the number of DNA patterns (i.e., the number of data sets to be processed). Each of the next *n* lines will contain a string of lower-case letters (at least one letter and at most 50 letters) that represent the pattern to check against the alien DNA grammar. Note that the input will be at least one letter even though the grammar allows for empty (null) string, i.e., empty string will not be in the input. Assume that input will not contain any character other than lower-case letters.

The Output:

For each DNA pattern, output the header "Pattern *i*: " where *i* is the number of the pattern in the input (starting with 1). Then, print "More aliens!" if the pattern matches the alien DNA description or "Still Looking." if the pattern does not match. Leave a blank line after the output for each data set. Follow the format illustrated in Sample Output.

(Sample Input/Output on the next page)

Sample Input:

```
2
aacbcbcc
aa
```

Sample Output:

```
Pattern 1: More aliens!

Pattern 2: Still Looking.
```