# UCF "Practice" Local Contest — August 27, 2011

## Common Patterns (filename: ngram)

An "n-gram" is a contiguous sub-sequence of n items from a given sequence. For example, given the sequence "ALIGAME", its only 5-grams are ALIGA, LIGAM and IGAME. There are special names for the first few n-grams: 1-gram is called unigram, 2-gram is called bigram (digram), and 3-gram is called trigram.

You are to write a program that, given a paragraph, will find the most-frequently appearing unigram, bigram and trigram. We are interested in n-grams consisting of letters only. More specifically, you are to find the single letter that appears the most, the two consecutive letters that appear the most, and the three consecutive letters that appear the most. If there is more than one candidate for a given subsequence (e.g., several bigrams appearing the most), print the one that comes first alphabetically (i.e., smallest when compared as strings).

Note that "consecutive" letters means one letter immediately after another letter, i.e., no other characters (spaces or other separators) in between.

### *The Input:*

There will be multiple data sets (paragraphs) to be processed. The first input line for a data set (paragraph) is an integer $p$ ($1 \leq p \leq 50$) indicating the number of lines in the paragraph. The following $p$ input lines provide the text (contents) for the paragraph. Each of these input lines will contain only lowercase letters, spaces, commas and periods. Assume that these input lines will not exceed column 70 and that each line will contain at least one letter. (Note that the only separators are spaces, commas, periods, and end-of-line.) End of data is indicated by a value of zero for $p$ (number of lines for a paragraph).

### *The Output:*

Print a heading for each paragraph, followed by its most-frequently appearing unigram, bigram and trigram (assume that each input paragraph will contain answers for each of these). Leave a blank line after the output for each data set. Follow the format illustrated in Sample Output.

Note that for a string such as "aaaaaa", some interpretations view it as having three copies of "aa" and some view it as having five occurrences of "aa". Use the latter view for this problem (same concept applies to trigrams as well).

(Sample Input/Output on the next page)

## *Sample Input:*

```
4
z z. z,z.  z z. z,z.
go  go  go
go  go  go
ali  ali   ali
3
a a. a,a.
bc    bc
abcd   abcd   abcd
0
```

## *Sample Output:*

```
Paragraph #1:
   Unigram: z
   Bigram:  go
   Trigram: ali

Paragraph #2:
   Unigram: a
   Bigram:  bc
   Trigram: abc
```