



Ashish Agrawal <ashish.agrawal@payu.in>

Another ruby problem

Ashutosh Tiwari1 <ashutosh.tiwari1@payu.in>

Tue, Feb 10, 2015 at 11:31 AM

To: Ashish Agrawal <ashish.agrawal@payu.in>, Divyanshu Verma <divyanshu.verma@payu.in>

Computing "subword" combinations.

Accept any number of lines of input entered from standard input. A valid line contains a word and a number. The word comes first, and is up to twenty alphabetic characters (A-Z) in upper or lower case. The word may be preceded and followed by any number of blanks. The number is a decimal integer greater than zero, and may be followed by any number of blanks or a return. Input is terminated by EOF character. For each input word, print all possible "subwords" that have as many characters as specified by the number. A "subword" is a n-letter word comprised of only characters that are found in the given word. If you are familiar with the game Scrabble, it's as though the input word is your pool of letters, and you want to find all possible arrangements of n letters from your pool. .nf For example, CAT 2 would produce CA CT AC AT TA TC and DOG 1 would produce D O G and BIRD 3 would produce BIR BID IRD RID DIR RIB ... Note, that we are producing combinations WITHOUT replacement, so in the BIRD 3 example, BBB and RRR are NOT possible subwords. The output should echo the input line to standard output, followed by the list of subwords, one word per line, left justified. Each set of output words should be followed by one blank line. The program should respond to invalid input by echoing the input line to the output, and then issuing a one line message indicating the nature of the error:

"Illegal character in word"

"Illegal character in number"

"Number greater than length of word"

"Word longer than 20 characters"

After issuing the message, processing should resume with the next input line. When all input has been processed, issue a message indicating the number of input words that were processed (including word with errors) e.g., "End of input encountered, 4 words processed."