TYPE AHEAD

CHALLENGE DESCRIPTION:

Your task is to build a type-ahead feature for an upcoming product.

When the user enters a word or set of words, we want to be able to "predict" what they're going to type next with some level of accuracy. We've chosen to implement this using the N-Gram algorithm defined here.

Your program should return a tuple of predictions sorted high to low based on the prediction score (upto a maximum of three decimal places, or pad with zeroes upto three decimal places i.e. 0.2 should be shown as 0.200), (if predictions share the same score, they are sorted alphabetically.) Words should be split by whitespace with all non-alphanumeric characters stripped off the beginning and end.
Prediction scores are calculated like this: Occurrences of a word after an N-gram / total number of words after an N-gram e.g. for an N-gram of length 2:

```
ONE TWO ONE TWO THREE TWO THREE
```

"TWO" has the following predictions:

```
THREE:0.666,ONE:0.333
```

"THREE" occurred 2 times after a "TWO" and "ONE" occurred 1 time after a "TWO", for a total of 3 occurrences after a "TWO".

Your program will run against the following text, ignoring all punctuation i.e. Hardcode it into your program:

```
Mary had a little lamb its fleece was white as snow;
And everywhere that Mary went, the lamb was sure to go.
It followed her to school one day, which was against the rule;
It made the children laugh and play, to see a lamb at school.
And so the teacher turned it out, but still it lingered near,
And waited patiently about till Mary did appear.
"Why does the lamb love Mary so?" the eager children cry; "Why, Mary loves the lamb, you know" the teacher did reply."
```

INPUT SAMPLE:

Your program should accept as its first argument a path to a filename.The input file contains several lines. Each line is one test case. Each line contains a number followed by a string, separated by a comma. E.g.

```
2,the
```

The first number is the n-gram length. The second string is the text printed by the user and whose prediction you have to print out.

OUTPUT SAMPLE:

For each set of input produce a single line of output which is the predictions for what the user is going to type next. E.g.

```
lamb,0.375;teacher,0.250;children,0.125;eager,0.125;rule,0.125
```