Natural Language Processing is the sub field of artificial intelligence that deals with designing algorithms, programs, and systems that can understand human languages in written and spoken forms. A useful initial analysis in NLP is the extraction of n-grams, that is, sequences of n words. These n-grams commonly encode concepts and are used by several natural language processing algorithms.

Consider for example the first paragraph in this assignment. Each individual word constitutes a 1-gram. The 2-grams are: "Natural Language", "Language Processing", "Processing is", "is the", and so on. The first 3-gram is "Natural Language Processing", et cetera.

Your task is to implement a very simple text analysis tool that will receive a text file and integer n and k and will print all n-grams in the text that appear at least k times. Notice that the number of possible n-grams is huge (actually \( |V|^n \), where V is the vocabulary), so an array of counters is unfeasible. Instead of an array of counters, use a hash table that solves collisions by chaining to keep track of the n-grams that have been read. Modify the iNode object described in class to contain the n-gram, encoded as a single string and a counter for the number of occurrences. Since the key used for hashing is a string, you need to convert it to an integer value in a way that would ultimately result in as few collisions as possible. A simple way is to add the int values of all the characters in the strings, and then apply the mod operation, but perhaps you can propose a method that results in fewer collisions. Also, we will consider strings in upper and lowercase as being equivalent, so convert your strings to lowercase before hashing.

Run experiments with several plain text files and different values of n and k and write a report describing your results. Include at least some of your input text files in the electronic but not in the printed version of the report.