```
 1: // Copyright 2022 Anson Cheang, and Andy Nguyen
 2: #ifndef _HOME_IIFORCE_BADNAME_COMP4_PS6_RANDWRITER_H_
 3: #define _HOME_IIFORCE_BADNAME_COMP4_PS6_RANDWRITER_H_
 4:
 5: #include <iostream>
 6: #include <string>
 7: #include <map>
 8:
 9: using namespace std; //NOLINT
10:
11: class RandWriter {
12:  public:
13: // create a Markov model of order k from given text
14: RandWriter(string text, int k);  // Assume that text has length at least
k.
15: // ------------------------------------------------------------------
16: int orderK() const;  // order k of Markov model
17: // ------------------------------------------------------------------
18: // number of occurrences of kgram in text
19: int freq(string kgram) const;  // throw an exception if kgram is not of
20: // length k
21: // ------------------------------------------------------------------
22: // number of times that character c follows kgram
23: // if order=0, return num of times char c appears
24: // (throw an exception if kgram is not of length k)
25: int freq(string kgram, char c) const;
26: // ------------------------------------------------------------------
27: // random character following given kgram
28: // (Throw an exception if kgram is not of length k.
29: // Throw an exception if no such kgram.)
30: char kRand(string kgram);
31: // ------------------------------------------------------------------
32: // generate a string of length L characters
33: // by simulating a trajectory through the corresponding
34: // Markov chain. The first k characters of the newly
35: // generated string should be the argument kgram.
36: // Throw an exception if kgram is not of length k.
37: // Assume that L is at least k.
38: string generate(string kgram, int L);
39: // ------------------------------------------------------------------
40: friend ostream& operator<<(ostream& out, RandWriter& markov);
41: // overload the stream insertion operator and display
42: // the internal state of the Markov Model. Print out
43: // the order, the alphabet, and the frequencies of
44: // the k-grams and k+1-grams.
45:
46:  private:
47: string alphabet; //NOLINT
48: int order;
49: map<string, map<char, int>> RM;
50: };
51:
52: ostream& operator<<(ostream& out, RandWriter& markov);
53:
54: #endif  // _HOME_IIFORCE_BADNAME_COMP4_PS6_RANDWRITER_H_
```