```
 1: /* <Copyright Abara Mehene*/
 2: #ifndef MARKOV_MODEL_HPP
 3: #define MARKOV_MODEL_HPP
 4:
 5: #include <iostream>
 6: #include <map>
 7: #include <string>
 8: #include <stdexcept>
 9: #include <algorithm>
10:
11: class MarkovModel {
12:  private:
13:     int _order;
14:     std::map <std::string, int> _kgrams;  // must #include <map>
15:     std::string _alphabet;
16:     // space
17:  public:
18:     // create a Markov model of order k from given text
19:     // Assume that text has length at least k.
20:     MarkovModel(std::string text, int k);
21:
22:     // order k of Markov model
23:     int order();
24:
25:     // number of occurrences of kgram in text
26:     // (throw an exception if kgram is not of length k)
27:     int freq(std::string kgram);
28:
29:     // number of times that character c follows kgram
30:     // if order=0, return num of times char c appears
31:     // (throw an exception if kgram is not of length k)
32:     int freq(std::string kgram, char c);
33:
34:     // random character following given kgram
35:     // (Throw an exception if kgram is not of length k.
36:     //  Throw an exception if no such kgram.)
37:     char randk(std::string kgram);
38:
39:     // generate a string of length T characters
40:     // by simulating a trajectory through the corresponding
41:     // Markov chain.  The first k characters of the newly
42:     // generated string should be the argument kgram.
43:     // Throw an exception if kgram is not of length k.
44:     // Assume that T is at least k.
45:     std::string gen(std::string kgram, int T);
46:
47:     // overload the stream insertion operator and display
48:     // the internal state of the Markov Model. Print out
49:     // the order, the alphabet, and the frequencies of
50:     // the k-grams and k+1-grams.
51:     friend std::ostream& operator<<(std::ostream &out, MarkovModel &mm);
52:
53:   ~MarkovModel();
54: };
55:
56: #endif
```