```
    1: // Copyright 2022 Matthew Lorette Anaya
    2: #include "RandWriter.h"
    3:
    4: #define BOOST_TEST_DYN_LINK
    5: #define BOOST_TEST_MODULE Main
    6: #include <boost/test/unit_test.hpp>
    7:
    8: BOOST_AUTO_TEST_CASE(base_test) {
    9:     std::cout << "******************** Test Case 1 ********************"
<<
   10:     std::endl;
   11:
   12:     int k = 2;
   13:     std::string str = "gagggagaggcgagaaa";
   14:     RandWriter rw(str, k);
   15:
   16:     std::cout << "Printing out Markov Table for string:\n" <<
   17:     str << std::endl << std::endl;
   18:     std::cout << rw << std::endl;
   19:
   20:     std::cout << "Testing orderK and freq functions" << std::endl;
   21:     BOOST_REQUIRE(rw.orderK() == k);
   22:     BOOST_REQUIRE(rw.freq("gg") == 3);
   23:     BOOST_REQUIRE(rw.freq("ga", 'g') == 4);
   24:
   25:     std::cout << "Testing kRand function" << std::endl;
   26:     char rand = rw.kRand("aa");
   27:     BOOST_REQUIRE(rand == 'a' || rand == 'g');
   28:
   29:     std::cout << "Testing generate function" << std::endl << std::endl;
   30:     BOOST_REQUIRE(rw.generate("ga", 10).length() == 10);
   31: }
   32:
   33: BOOST_AUTO_TEST_CASE(exception_test) {
   34:     std::cout << "******************** Test Case 2 ********************"
<<
   35:     std::endl;
   36:     std::cout << "Testing construction exception: RandWriter('ADF', 4)" <
<
   37:     std::endl;
   38:
   39:     BOOST_REQUIRE_THROW(RandWriter("ADF", 4), std::invalid_argument);
   40:
   41:     std::cout << "Testing function exceptions" << std::endl;
   42:     RandWriter testMM("abc", 3);
   43:     BOOST_REQUIRE_THROW(testMM.freq("a"), std::runtime_error);
   44:     BOOST_REQUIRE_THROW(testMM.freq("ab", 'b'), std::runtime_error);
   45:     BOOST_REQUIRE_THROW(testMM.kRand("g"), std::runtime_error);
   46: }
```