

```
1: CC = g++
2: CFLAGS = -Wall -Werror -pedantic --std=c++14
3: LIBS = -lboost_unit_test_framework -lboost_date_time -lboost_regex
4: DEPS = RandWriter.h
5: SFMLFLAGS = -lsfml-graphics -lsfml-window -lsfml-system -lsfml-audio
6:
7: %.o: %.cpp $(DEPS)
8:     $(CC) $(CFLAGS) -c $<
9:
10: all: ps7
11:
12: ps7: KronosTime.o
13:     $(CC) -g $(CFLAGS) -o ps7 $^ $(LIBS) $(SFMLFLAGS)
14:     cpplint --filter=--runtime/references *.cpp
15:
16: clean:
17:     rm *.o ps7
```

```
1: // date and time sample code
2: // Copyright (C) 2015 Fred Martin
3: // Tue Apr 21 17:37:46 2015
4:
5: // compile with
6: // g++ datetime.cpp -lboost_date_time
7: // Y. Rykalova 4/12/2021
8:
9: // http://www.boost.org/doc/libs/1_58_0/doc/html/date_time/gregorian.html
10: // http://www.boost.org/doc/libs/1_58_0/doc/html/date_time/posix_time.htm
```

1

```
11:
12: #include <iostream>
13: #include <string>
14: #include "boost/date_time/gregorian/gregorian.hpp"
15: #include "boost/date_time/posix_time/posix_time.hpp"
16:
17: using std::cout;
18: using std::cin;
19: using std::endl;
20: using std::string;
21:
22: using boost::gregorian::date;
23: using boost::gregorian::from_simple_string;
24: using boost::gregorian::date_period;
25: using boost::gregorian::date_duration;
26:
27: using boost::posix_time::ptime;
28: using boost::posix_time::time_duration;
29:
30: int main() {
31:     // Gregorian date stuff
32:     string s("2015-01-01");
33:     date d1(from_simple_string(s));
34:     date d2(2015, boost::gregorian::Apr, 21);
35:
36:     date_period dp(d1, d2); // d2 minus d1
37:
38:     date_duration dd = dp.length();
39:
40:     cout << "duration in days " << dd.days() << endl;
41:
42:     // Posix date stuff
43:     ptime t1(d1, time_duration(0, 0, 0, 0)); // hours, min, secs, nanosecs
44:     ptime t2(d2, time_duration(0, 0, 0, 0));
45:
46:     time_duration td = t2 - t1;
47:
48:     cout << "duration in hours " << td.hours() << endl;
49:     cout << "duration in ms " << td.total_milliseconds() << endl;
50: }
```