```
 1: // $Id: division.cpp,v 1.6 2014-05-29 19:11:29-07 - - $
 2:
 3: #include <cstdlib>
 4: #include <iostream>
 5: #include <stdexcept>
 6:
 7: using namespace std;
 8:
 9: using ulong = unsigned long;
10: using uupair = pair<ulong,ulong>;
11:
12: uupair divide (const ulong &dividend, const ulong &divisor) {
13:    if (divisor == 0) throw domain_error ("divide(_,0)");
14:    ulong powerof2 = 1;
15:    ulong divisor_ = divisor;
16:    while (divisor_ < dividend) {
17:       divisor_ *= 2;
18:       powerof2 *= 2;
19:    }
20:    ulong quotient = 0;
21:    ulong remainder = dividend;
22:    while (powerof2 > 0) {
23:       if (divisor_ <= remainder) {
24:          remainder -= divisor_;
25:          quotient += powerof2;
26:       }
27:       divisor_ /= 2;
28:       powerof2 /= 2;
29:    }
30:    return uupair (quotient, remainder);
31: }
32:
```

```
33:
34: ostream &operator<< (ostream &out, const uupair &pair) {
35:    out << pair.first << " R " << pair.second;
36:    return out;
37: }
38:
39: uupair tests[] = {
40:    {             0L, 1024L},
41:    {             5L,    7L},
42:    {           100L,    0L},
43:    {           100L,   50L},
44:    {           320L,   20L},
45:    {           963L,   71L},
46:    {12345678912345L, 9876L},
47: };
48:
49: int main (int argc, char **argv) {
50:    (void) argc; // warning: unused parameter 'argc'
51:    (void) argv; // warning: unused parameter 'argv'
52:    uupair *testend = tests + sizeof tests / sizeof *tests;
53:    for (uupair *itor = tests; itor < testend; ++itor) {
54:       ulong dividend = itor->first;
55:       ulong divisor = itor->second;
56:       cout << dividend << " / " << divisor << " = ";
57:       try {
58:          uupair result = divide (dividend, divisor);
59:          cout << result;
60:          uupair tested = uupair (dividend / divisor,
61:                                  dividend % divisor);
62:          if (tested != result) {
63:             cout << ": wrong " << tested;
64:          }
65:       }catch (domain_error &error) {
66:          cout << "domain_error: " << error.what();
67:       }
68:       cout << endl;
69:    }
70:    return EXIT_SUCCESS;
71: }
72:
73: //TEST// ./division 2>&1 >division.output
74: //TEST// mkpspdf division.ps division.cpp* division.output
75:
```

```
   1: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: starting division.cpp
   2: division.cpp:
   3:      $Id: division.cpp,v 1.6 2014-05-29 19:11:29-07 - - $
   4: g++ -g -O0 -Wall -Wextra -rdynamic -std=gnu++11 division.cpp -o division
-lglut -lGLU -lGL -lX11 -lrt -lm
   5: rm -f division.o
   6: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: finished division.cpp
```

```
1: 0 / 1024 = 0 R 0
2: 5 / 7 = 0 R 5
3: 100 / 0 = domain_error: divide(_,0)
4: 100 / 50 = 2 R 0
5: 320 / 20 = 16 R 0
6: 963 / 71 = 13 R 40
7: 12345678912345 / 9876 = 1250068743 R 6477
```