

```
1: // $Id: division.cpp,v 1.9 2016-03-23 18:39:53-07 - - $
2:
3: #include <cstdlib>
4: #include <iostream>
5: #include <locale>
6: #include <stdexcept>
7:
8: using namespace std;
9:
10: using ulong = unsigned long;
11: using uupair = pair<ulong,ulong>;
12:
13: uupair divide (const ulong &dividend, const ulong &divisor) {
14:     if (divisor == 0) throw domain_error ("divide(_,0)");
15:     ulong powerof2 = 1;
16:     ulong divisor_ = divisor;
17:     while (divisor_ < dividend) {
18:         divisor_ *= 2;
19:         powerof2 *= 2;
20:     }
21:     ulong quotient = 0;
22:     ulong remainder = dividend;
23:     while (powerof2 > 0) {
24:         if (divisor_ <= remainder) {
25:             remainder -= divisor_;
26:             quotient += powerof2;
27:         }
28:         divisor_ /= 2;
29:         powerof2 /= 2;
30:     }
31:     return uupair (quotient, remainder);
32: }
33:
```

```
34:
35: ostream &operator<< (ostream &out, const uupair &pair) {
36:     out << pair.first << " Rem " << pair.second;
37:     return out;
38: }
39:
40: uupair tests[] = {
41:     {          0L, 1'024L},
42:     {          5L,   7L},
43:     {        100L,   0L},
44:     {        100L,  50L},
45:     {        320L,  20L},
46:     {        963L,  71L},
47:     {12'345'678'912'345L, 9'876L},
48: };
49:
50: int main (int argc, char **argv) {
51:     (void) argc; // warning: unused parameter 'argc'
52:     (void) argv; // warning: unused parameter 'argv'
53:     cout.imbue (locale (""));
54:     uupair *testend = tests + sizeof tests / sizeof *tests;
55:     for (uupair *itor = tests; itor < testend; ++itor) {
56:         ulong dividend = itor->first;
57:         ulong divisor = itor->second;
58:         cout << dividend << " / " << divisor << " = ";
59:         try {
60:             uupair result = divide (dividend, divisor);
61:             cout << result;
62:             uupair tested = uupair (dividend / divisor,
63:                                     dividend % divisor);
64:             if (tested != result) {
65:                 cout << ": wrong " << tested;
66:             }
67:         } catch (domain_error &error) {
68:             cout << "domain_error: " << error.what();
69:         }
70:         cout << endl;
71:     }
72:     return EXIT_SUCCESS;
73: }
74:
75: //TEST// ./division 2>&1 >division.output
76: //TEST// mkpspdf division.ps division.cpp* division.output
77:
```

[illegible]

```
1: 0 / 1,024 = 0 Rem 0
2: 5 / 7 = 0 Rem 5
3: 100 / 0 = domain_error: divide(_,0)
4: 100 / 50 = 2 Rem 0
5: 320 / 20 = 16 Rem 0
6: 963 / 71 = 13 Rem 40
7: 12,345,678,912,345 / 9,876 = 1,250,068,743 Rem 6,477
```