

```
1 #include <lnv3/lnv3.h>
2 #include <nttl/randomPrime.h>
3 #include <nttl/gcd.h>
4 #include <nttl/inverse.h>
5 #include <nttl/gcdEuclid.h>
6 #include <nttl/sqrt.h>
7
8 void bubbleSort1(ln x[], int n);
9
10 #define SHIFT 1000
11
12 void main()
13 {
14     ln N =
15         "5600841207178799403662441921274579265855140375130798595561073254468043003990757003
16         76144191028941575169525004256074543693069441058182779623767774475999446585111678248
17         6081";
18     ln sqrtN =
19         "2366609644022182083963051296605304336438037353230565934003185921678149657166148220
20         048";
21     ln decimals = Sqrt( N * ln( 10 ).FastExp( 2 * SHIFT, N ) ) % ln( 10 ).FastExp(
22         SHIFT, N );
23     // Value returned as lowest:
24     //
25     16737891118280003723066712066117277274725556063785788062480325601562046241336572305
26     280
27     // The previous lowest value (with the bad algorithm) was:
28     //
29     47332192880443641679261025932106086728760747064611318680063718433562993143322964400
30     97
31
32     ln list[500];
33
34     int i;
35     ln prevx = ln( -1 );
36     for( i = 0; i < 500; i++ )
37     {
38         ln a = decimals + 0;
39         ln b = ln( 10 ).FastExp( SHIFT, N );
40         //cout << a << endl;
41         //cout << b << endl;
42         ln oa = ln();
43         ln ob = ln();
44         for( int f = 0; f < i; f++ )
45         {
46             oa = ln();
47             ob = ln();
48             EGCD( a, b, &oa, &ob );
49             //cout << "oa=" << oa << ",ob=" << ob << endl;
50             if( ( oa == 1 ) || ( ob == 1 ) )
51                 break;
52             a = oa.Abs();
53             b = ob.Abs();
54             oa = oa.Abs();
55             ob = ob.Abs();
56         }
57         if( oa < ob )
58         {
59             ln t = ob;
60             ob = oa;
61             oa = t;
62         }
63         ln x = ( ob * sqrtN + oa ) % N;
64         ln y = ( x * x ) % N;
65     }
66 }
```

```
58
59     if( x == prevx )
60         break;
61
62     //cout << oa << ", " << ob << " = " << endl << x << endl << ", x^2 = " << y <<
endl;
63
64     if( ( x > sqrtN ) && ( x < ( N - sqrtN ) ) &&
65         ( x > 0 ) )
66     {
67         //cout << x << endl;
68         //cout << y << endl;
69         //cout << "---" << endl;
70         list[ i ] = y;
71     }
72
73     prevx = x.Abs();
74 }
75
76 bubbleSort1( list, 500 );
77 int n = 0;
78 while( list[ n ] == 0 )
79     n++;
80 cout << list[ n ] << endl;
81
82 return;
83 }
84
85 void bubbleSort1(ln x[], int n) {
86     for (int pass=1; pass < n; pass++) {
87         for (int i=0; i < n-pass; i++) {
88             if (x[i] > x[i+1]) {
89                 ln temp = x[i]; x[i] = x[i+1]; x[i+1] = temp;
90             }
91         }
92     }
93 }
```