

```
#include <iostream>
#include <string>

using namespace std;

int main (int argc, char * const argv[])
{
    int lengths[ 739 ];
    bool current[ 739 ];

    int longest = 0;

    for( int n = 0; n < 739; n++ )
    {
        memset( current, false, 739 * sizeof( bool ) );

        int index = n;
        int length = 0;

        while( true )
        {
            if( current[ index ] == true )
            {
                break;
            }
            else
            {
                current[ index ] = true;
                length++;
                index = ( index * index + 1 ) % 739;
            }
        }

        // remove last one (the repetition)
        length--;

        lengths[ n ] = length;
        if( length > longest )
            longest = length;
    }

    cout << "Longest=" << longest << endl;

    int i = 0;
    for( int n = 0; n < 739; n++ )
    {
        if( lengths[ n ] == longest )
        {
            i++;
        }
    }
}
```

```
    cout << endl << "Path " << i << ": ";
    int p = n;
    for( int x = 0; x < longest; x++ )
    {
        cout << p << ( ( x == longest - 1 ) ? " " : ", " );
        p = ( p * p + 1 ) % 739;
    }
}

return 0;
}
```

```
/*
```

Results:

Longest=37

Path 1: 256, 505, 71, 608, 165, 622, 388, 528, 182, 609, 643, 349, 606, 693, 639, 394,  
47, 732, 50, 284, 106, 152, 196, 728, 122, 105, 680, 526, 291, 436, 174, 717, 485,  
224, 664, 453, 507

Path 2: 319, 519, 366, 198, 38, 706, 351, 528, 182, 609, 643, 349, 606, 693, 639, 394,  
47, 732, 50, 284, 106, 152, 196, 728, 122, 105, 680, 526, 291, 436, 174, 717, 485,  
224, 664, 453, 507

Path 3: 420, 519, 366, 198, 38, 706, 351, 528, 182, 609, 643, 349, 606, 693, 639, 394,  
47, 732, 50, 284, 106, 152, 196, 728, 122, 105, 680, 526, 291, 436, 174, 717, 485,  
224, 664, 453, 507

Path 4: 483, 505, 71, 608, 165, 622, 388, 528, 182, 609, 643, 349, 606, 693, 639, 394,  
47, 732, 50, 284, 106, 152, 196, 728, 122, 105, 680, 526, 291, 436, 174, 717, 485,  
224, 664, 453, 507

```
*/
```