

Example

find $3^{63} \bmod 160$

so:
a=3
b=63
c=160

write 63 as binary
111111

111111 has 6 digits

$\text{sqm}[0] = 3 \bmod 160 = 3$

loop:
 $\text{sqm}[1] = 3^2 \bmod 160 = 9$
 $\text{sqm}[2] = 9^2 \bmod 160 = 81$
 $\text{sqm}[3] = 81^2 \bmod 160 = 1$
 $\text{sqm}[4] = 1^2 \bmod 160 = 1$
 $\text{sqm}[5] = 1^2 \bmod 160 = 1$

$\text{answer} = 3^1 = 3$

loop:
 $\text{answer} = (3 \times 9)^1 \bmod 160 = 27$
 $\dots = (27 \times 81)^1 \bmod 160 = 107$
 $\dots = (107 \times 1)^1 \bmod 160 = 107$
 $\dots = (107 \times 1)^1 \bmod 160 = 107$
 $\dots = (107 \times 1)^1 \bmod 160 = 107$

So answer is 107

Method

```
static int AtoBmodC(int a, int b, int c) {  
  
    List<int> bin = new List<int>();  
    List<int> sqm = new List<int>();  
    int i, answer;  
  
    // List bin will contain digits of B written as binary  
  
    while(b != 0) {  
        bin.Add(b % 2);    // gives either a 0 or 1  
        b /= 2;           // divide by 2 and truncate  
    }  
  
    // Binary number has a length of bin.Count  
    // List sqm will contain A raised to powers of two, mod C  
  
    sqm.Add(a % c);  
    for (i = 1; i < bin.Count; i++)  
        sqm.Add((sqm[i-1] * sqm[i-1]) % c);  
  
    // Compute the answer by multiplying together the proper  
    // powers of two, mod C  
  
    answer = Math.Pow(sqm[0],bin[0]);  
    for (i = 1; i < bin.Count; i++)  
        answer = Math.Pow(answer * sqm[i] , bin[i]) % c;  
  
    return answer;  
  
} // from krellinst.org
```