Intercept

Long long ago, so long ago, no body knows how long ago, there was a huge galactic war. There was a very powerful general, General Ramuk, who had every possible soldier and scientist under him. One of his scientists reports that he had intercepted a transmission that he believes is from the aliens. A group of experienced cryptographers believe that in the following hypotheses:

1. The aliens follow use binary system for representing numbers
2. Their '0' should be interpreted as '1' and vice-versa
3. The message is encoded as follows: [32bits of n1][16bits of n2][n2 bits of n3]
4. The retransmission they expect is: [32bits of the remainder] when **n3** is divided by **n1**.
5. All the numbers are written, Most significant bit first.
6. Remainder must be communicated in the following format:
   [remainder for 1st instance]
   [remainder for 2nd instance]
   in their own number system, without leading '1's ('0's).
7. The number of instances is about **200**.

The first transmission was completed. Ramuk is eagerly waiting for the second transmission, which must be replied. Being such a simple problem, he asks you to write a program to do the same.

He says: *"Nee evalovu chinnadha codea ezhudhariyo avalovu parisu onnakku kaathhirriku "*, which translates to: *"The smaller the code you write, that much reward is awaiting you..."*.

You want to save the world from a probable Alien Invasion, and get as much money as possible.

Constant bit length numbers will be prefixed by '1's ('0's in their notation).

Scoring

The scoring for this problem is the **length of the source code**.

Sample Input

NOTE: The colons (:) and newlines are for clarity

```
11111111111111111111110001001110:1111111111101111:0011100111000100
11111111111111111111110001001011:1111111111101100:0100100001000011111
```

The actual input will be like:

```
1111111111111111111111000100111011111111111101111001110011100010
111111111111111111111100010010111111111111110110001001000010000111111
```

(new line is again, for clarity)

Sample Output

```
0101101001
0010001111
```

Explanation

n1=945
n2=16
n3=50747
output=662

n1=948
n2=20
n3=376288
output=880

Warning: Large Input.