Little Sander found a lot of numbers lying around, so he decided to have some fun with them. Since he's a #1 fan of prime numbers, he decided to associate each number he found with a prime number. He's also extremely proficient in the binary numeral system, so it's no wonder his association technique has something to do with it.

For each number num Sander found, he calculates the number of '1's in its binary representation. Let this number be o . The prime number associated with num is thus the othprime number (0-based).

Given num, your task is to find the prime number associated with it. You should probably keep in mind that Sander believes that 0 and 1 are also prime numbers, and we're not going to argue with him right now.

**Example**

For num = 15, the output should be  
onesPosition(num) = 5.

1510 = 11112, so the answer is the fourth 0-based prime number (as Sanders sees them). Here are the first five Sanders-prime numbers: 0, 1, 2, 3, 5. Thus, the output should be 5.

**Input/Output**

* **[time limit] 3000ms (cs)**
* **[input] integer num**

*Guaranteed constraints:*  
0 ≤ k < 231.

* **[output] integer**

The prime number that Sander associates with num.

<https://codefights.com/challenge/QKHjjtxSpyyyuEGXN?utm_source=emailNotification&utm_medium=email&utm_campaign=featuredChallenge>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static int onesPosition(int num)

{

int max = 1000;

bool[] prime = Enumerable.Repeat(true, max+1).ToArray();

for (int p = 2; p \* p <= max; p++)

{

// If prime[p] is not changed, then it is a prime

if (prime[p] == true)

{

// Update all multiples of p

for (int i = p \* 2; i <= max; i += p)

prime[i] = false;

}

}

List<int> lp = new List<int>();

//lp.Add(0);

//lp.Add(1);

for (int p = 0; p <= max; p++)

if (prime[p])

lp.Add(p);

return lp[Convert.ToString(num, 2).Count(e => e == '1')];

}

static void Main(string[] args)

{

Console.WriteLine( onesPosition(107));

Console.ReadLine();

}

}

}