A number is considered to be *boring* if:

* it's a positive integer;
* it's not prime;
* it does not belong to the Fibonacci sequence.

Given a number k, find the kth *boring* number when counting up from 1.

**Example**

* For k = 1, the output should be  
  kthBoring(k) = 4.

The numbers 1, 2 and 3 belong to the Fibonacci sequence, so they are not *boring*. The number 4 is not prime and does not belong to the Fibonacci sequence, so it's the 1st *boring* number.

* For k = 2, the output should be  
  kthBoring(k) = 6.

The number 5 is prime, so it's not *boring*, meaning that the 2nd *boring* number is 6.

**Input/Output**

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

*Constraints:*  
1 ≤ k ≤ 2 · 105.

* **[output] integer**

The kth *boring* number when counting up from 1.

<https://codefights.com/challenge/2ST7pD433eT8SXNyM?utm_source=featuredChallenge&utm_medium=email&utm_campaign=email_notification>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.IO;

namespace ConsoleApplication1

{

class Program

{

static bool esPrimo(int n)

{

if (n < 2) return false;

if (n == 2) return true;

if (n % 2 == 0) return false;

int sqr = (int)Math.Sqrt(n);

for (int i = 3; i <= sqr; i += 2)

{

if (n % i == 0) return false;

}

return true;

}

static bool esFib(int n)

{

int[] fib = { 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 75025, 121393, 196418, 317811 };

int indice = Array.BinarySearch(fib, n);

return (indice > -1) ? true : false;

}

static int kthBoring(int k)

{

int kth = 0;

int i =0;

for ( i = 1; ; i++)

{

if (!esPrimo(i) && !esFib(i))

{

kth++;

}

if (kth == k)

{

return i;

}

}

}

static void Main(string[] args)

{

//Console.WriteLine(esFib(9));

Console.WriteLine(kthBoring(1));

Console.ReadLine();

}

}

}