Given a non-negative integer n, your task is to calculate the sum of its non-prime divisors.

**Example**

For number = 8, the output should be  
sumDivisors(number) = 13.

number's divisors are 1, 2, 4 and 8. 2 is a prime, so it shouldn't be added to the result. The answer is thus 1 + 4 + 8 = 13.

**Input/Output**

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

*Cconstraints:*  
0 ≤ number ≤ 107

* **[output] integer**

<https://codefights.com/challenge/A9hB67Kfqc7ADHGCo>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

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 int sumDivisors(int number)

{

int sum = 0;

for (int i = 1; i <= number / 2; i++)

{

if (number %i==0 && !esPrimo(i))

{

sum += i;

}

}

if (!esPrimo(number))

{

sum += number;

}

return sum;

}

static void Main(string[] args)

{

Console.WriteLine(sumDivisors(9));

//for (int i = 1; i < 100; i++)

//{

// Console.Write(esPrimo(i) ? i + " " : "");

//}

Console.ReadLine();

}

}

}