A *pronic* number (also known as *oblong* number,*rectangular* number or *heteromecic* number), is a number which is the product of two consecutive integers, i.e. it can be represented as x \* (x + 1) for some integer x. The first few *pronic*numbers are:

0, 2, 6, 12, 20, 30, 42...

Here's why:

* 0 = 0 × 1
* 2 = 1 × 2
* 6 = 2 × 3
* 12 = 3 × 4
* 20 = 4 × 5
* 30 = 5 × 6
* 42 = 6 × 7

Your task is to write a function isPronic which checks whether the number n is a *pronic*number and returns true if it is and falseotherwise.

**Example**

* For n = 0, the output should be  
  isPronic(n) = true.
* For n = 1, the output should be  
  isPronic(n) = false.

**Input/Output**

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

*Constraints:*  
0 ≤ n ≤ 225.

* **[output] boolean**

true if n is a *pronic* number, falseotherwise.

<https://codefights.com/challenge/qZfpoeALnuY26YNaf/main>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static bool isPronic(int n)

{

for (int i =0; i <= n; i++)

{

if (i \* (i + 1) ==n)

{

return true;

}

}

return false;

}

static void Main(string[] args)

{

Console.WriteLine(isPronic(1));

Console.ReadLine();

}

}

}

-----------otra solución--------------------

bool isPronic(int n)

{

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

return j \* j + j == n;

}