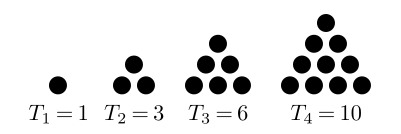
Author

[Kartik\_M](https://codefights.com/profile/Kartik_M)

https://codefights.com/img/coins_new.png2000

*Triangular numbers* are defined by the formula n \* (n + 1) / 2 with n starting from 1. They count the number of objects that can form an equilateral triangle as shown in the picture below:



So the sequence of *triangular numbers* begins as follows:  
1, 3, 6, 10, 15, 21, 28, ....

It is proven that the sum of squares of any two consecutive *triangular numbers* is equal to another*triangular number*.

In this challenge you're given a *triangular number* n. Return true if it can be represented as a sum of squares of two consecutive *triangular numbers*, orfalse otherwise.

**Example**

* For n = 6, the output should be  
  Triangular(n) = false.  
  No two squared consecutive *triangular numbers*add up to 6.
* For n = 45, the output should be  
  Triangular(n) = true.  
  3\*3 + 6\*6 = 9 + 36 = 45
* **[input] integer n**

A positive *triangular number*, 3 ≤ n ≤ 5 \* 107.

* **[output] boolean**

true if it is possible to represent n as the sum of squares of two consecutive *triangular numbers*, and false otherwise.

<https://codefights.com/challenge/MpK5nS6KiXtbLbkcm>

static bool Triangular(int n)

{

int t = 0, tsig =0;

for (int i = 1; ; i++)

{

//Console.WriteLine((i \* (i + 1)) / 2);

t = (i \* (i + 1)) / 2;

tsig = ((i+1) \* ((i+1) + 1)/2);

if (t \* t + tsig \* tsig == n)

{

return true;

}

if (t \* t + tsig \* tsig > n)

{

break;

}

}

return false;

}