**7 kyu**

**Excessively Abundant Numbers**

10394% of 86116 of810[RichAWarren](https://www.codewars.com/users/RichAWarren)

Java

* [TRAIN AGAIN](https://www.codewars.com/kata/excessively-abundant-numbers/train/java)
* [NEXT KATA](https://www.codewars.com/trainer/java)

Details

[Solutions](https://www.codewars.com/kata/excessively-abundant-numbers/solutions/java)

[Forks (3)](https://www.codewars.com/kata/excessively-abundant-numbers/forks/java)

[Discourse (11)](https://www.codewars.com/kata/excessively-abundant-numbers/discuss/java)

* Add to Collection
* |
* Share this kata:

An abundant number or excessive number is a number for which the sum of its proper divisors is greater than the number itself.

The integer 12 is the first abundant number. Its proper divisors are 1, 2, 3, 4 and 6 for a total of 16 (> 12).

Derive function abundantNumber(num)/abundant\_number(num) which returns true/True/.true. if num is abundant, false/False/.false. if not.

<https://www.codewars.com/kata/excessively-abundant-numbers/java>

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package javaapplication32;

public class JavaApplication32 {

static int SumDivisores(int n)

{

//List<int> div = new List<int>();

int sum = 0;

for (int i = 1; i \* i <= n; i++)

{

if (n % i == 0)

{

//div.Add(i);

sum += i;

if (n / i != i)

{

if (n / i != n)

{

//div.Add(n / i);

sum += (n / i);

}

}

}

}

return sum;

}

public static boolean abundantNumber(int num) {

// Your code goes here

return SumDivisores(num) > num;

}

public static void main(String[] args) {

// TODO code application logic here

System.out.println( abundantNumber(37));

}

}