

Euler 0012

The Problem:

What is the value of the first triangle number to have over five hundred divisors?

Considerations:

We don't actually need to compute all of the divisors of a number, since the number of divisors can be calculated by knowing the number of prime factors.

12, for example, has prime factors 2(2 of these) and 3. It has a total of 6 divisors, which can be calculated by taking the quantity of each prime factor and adding 1, 2 for 2 becomes 3 and 1 for 3 becomes 2... Ok, maybe that's an obtuse way of explaining it... If you want a good explanation of it, you can find it here: https://mathschallenge.net/library/number/number_of_divisors

The Approach:

We are going to use the same prime factors finder that we have before, but instead of we are going to multiple by a running divisors count.

The Code:

```
import math

#This is a modification of the prime factor finder
def calculate_divisors(number):
    upper_limit = number
    current_upper_limit = upper_limit
    sqrt_limit = math.sqrt(upper_limit)
    current_divisors = 1

    #we are going to hardcode 2 and 3 for looping sake
    if upper_limit % 2 == 0:
```

```

    current_div = 1
    while current_upper_limit % 2 == 0:
        current_upper_limit = current_upper_limit/2
        current_div += 1
    current_divisors *= current_div
if upper_limit % 3 == 0:
    current_div = 1
    while current_upper_limit % 3 == 0:
        current_upper_limit = current_upper_limit/3
        current_div += 1
    current_divisors *= current_div
current = 6

while current < sqrt_limit:
    #print(prime_factors, current_upper_limit)
    if current_upper_limit % (current - 1) == 0:
        current_div = 1
        while current_upper_limit % (current - 1) == 0:
            current_upper_limit = current_upper_limit/(current-1)
            current_div += 1
        current_divisors *= current_div
    if current_upper_limit % (current + 1) == 0:
        current_div = 1
        while current_upper_limit % (current + 1) == 0:
            current_upper_limit = current_upper_limit/(current+1)
            current_div += 1
        current_divisors *= current_div
    current += 6

return current_divisors

```

```

bailout = 100000 #keep checking up to a bailout number

```

```

triangle = 1
for i in range(2,bailout):
    triangle += i
    if calculate_divisors(triangle) >= 500:
        break

```

```
print(i, triangle)
```

Revision #2

Created 2025-08-26 15:46:14 UTC by Maxwell

Updated 2025-08-26 15:48:16 UTC by Maxwell