

Euler 0005

The Problem:

2520 is the smallest number evenly divisible by the numbers 1 through 10.

What is the smallest number evenly divisible by the numbers 1 through 20?

Considerations:

With the context provided we now know:

- That our function should produce 2520 with an upper limit of 10.
- The number we are looking for is less than the factorial of the upper limit.

The Approach:

One way we can approach this is to get the union of all the prime factors of each number.

There will be redundancy in this approach, but it should be performant enough.

Just listing out the first 6 numbers we should see:

- 2 [2]
- 3 [3]
- 4 [2,2]
- 5 [5]
- 6 [2,3]

Which should mean that the function of 6 would produce $2*2*3*5$ or 60.

I'll modify the prime factorizing method from Euler 3 to just produce prime factors.

The Code:

```
import math
```

```

smallest_multiple = []
upper_limit = 20

def prime_factors(upper_limit):
    #upper_limit = 20023110541 #, good to use to double check. Is [2,2,3,167]
    current_upper_limit = upper_limit
    sqrt_limit = math.sqrt(upper_limit)
    prime_factors = []

    #we are going to hardcode 2 and 3 for looping sake
    if upper_limit % 2 == 0:
        while current_upper_limit % 2 == 0:
            current_upper_limit = current_upper_limit/2
            prime_factors += [2]
    if upper_limit % 3 == 0:
        while current_upper_limit % 3 == 0:
            current_upper_limit = current_upper_limit/3
            prime_factors += [3]
    current = 6

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

    if current_upper_limit != 1:
        prime_factors += [current_upper_limit]

    return prime_factors

#start from 2 and go to the number we want to find
for i in range(2,upper_limit + 1,1):

```

```
current_pf = prime_factors(i)
new_smallest_multiple = []
#look at all of the numbers between our current set and what we just factorized
for number in set(smallest_multiple).union(set(current_pf)):
    #add that many prime factors to the current smallest multiple
    for j in range(max(smallest_multiple.count(number),current_pf.count(number))):
        new_smallest_multiple += [number]
    smallest_multiple = new_smallest_multiple

print(smallest_multiple)
print(math.prod(smallest_multiple))
```

Revision #2

Created 2025-07-01 19:03:54 UTC by Maxwell

Updated 2025-07-17 14:50:33 UTC by Maxwell