

Problem: Divisible by 7, not a multiple of 5

Write a program that returns a list of all numbers that are divisible by 7 but are not a multiple of 5, between 2000 and 3200 (both included).

The following problems are taken from <https://projecteuler.net> — an excellent website for practicing your problem solving techniques.

Problem 4: Largest Palindrome Product

A palindromic number reads the same both ways. The largest palindrome made from the product of two 2-digit numbers is $9009 = 91 \times 99$. Find the largest palindrome made from the product of two 3-digit numbers.

Problem 10: Summation of Primes

The sum of the primes below 10 is $2 + 3 + 5 + 7 = 17$. Find the sum of all the primes below two million.

Problem 20: Factorial Digit Sum

$n!$ means $n \times (n - 1) \times \cdots \times 3 \times 2 \times 1$. For example, $10! = 10 \times 9 \times \cdots \times 3 \times 2 \times 1 = 3628800$, and the sum of the digits in the number $10!$ is $3 + 6 + 2 + 8 + 8 + 0 + 0 = 27$. Find the sum of the digits in the number $100!$

Problem 39: Integer Right Triangles

If p is the perimeter of a right angle triangle with integral length sides, $\{a, b, c\}$, there are exactly three solutions for $p = 120$.

$$\{20, 48, 52\}, \{24, 45, 51\}, \{30, 40, 50\}$$

For which value of $p \leq 1000$, is the number of solutions maximized?

Sample Solutions

```
1 def div_7_not_5(low=2000, high=3200):
2     results = []
3     for i in range(low, high+1):
4         if i % 7 == 0 and i % 5 != 0:
5             results.append(i)
6
7     return results
8
9 def div_7_not_5_comp(low=2000, high=3200):
10    return [ i for i in range(low, high+1) if i % 7 == 0 and i % 5 != 0]
```

```
1 def largest_palindrome(low=100, high=999):
2     def is_palindrome(s):
3         """returns true if the string is the same forwards and backwards"""
4         return s == s[::-1]
5
6     pals = []
7     for i in range(low, high+1):
8         for j in range(i, high+1):
9             if is_palindrome(str(i*j)):
10                pals.append(i*j)
11    return max(pals)
12
13
14 def largest_palindrome_comp(low=100, high=999):
15    def is_palindrome(s):
16        """returns true if the string is the same forwards and backwards"""
17        return s == s[::-1]
18
19    return max([i*j for i in range(low, high+1) for j in range(i, high+1) if is_palindrome(str(i*j))])
```

```
1 def sum_of_primes(high=2000000):
2     def is_prime(n):
3         for i in range(2, int(math.sqrt(n))+1):
4             if n % i == 0:
5                 return False
6             return True
7
8     sum = 0
9     for i in range(2, high+1):
10        if is_prime(i):
11            sum += i
12        return sum
13
14
15 def sum_of_primes_comp(high=2000000):
16     def is_prime(n):
17         for i in range(2, int(math.sqrt(n))+1):
18             if n % i == 0:
19                 return False
20             return True
21
22     return sum([i for i in range(2, high+1) if is_prime(i)])
```