HackerRank

# Project Euler


Author : Avinash Sorab


Date Created : 24th Aug 2019

Language Used : Python 3

Platform : HackerRank

# Contents


1. **Multiples of 3 and 5**
2. **Even Fibonacci numbers**
3. **Largest prime factor**
4. **Largest palindrome product**
5. **Smallest multiple**
6. **Sum square difference**
7. **10001$^{st}$ prime**
8. **Largest product in a series**
9. **Special Pythagorean triplet**
10. **Summation of primes**
11. **Largest product in a grid**
12. **Highly divisible triangular number**
13. **Large sum**
14. **Longest Collatz sequence**
15. **Lattice paths**
16. **Power digit sum**
17. **Number to words**
18. **Maximum path sum I**
19. **Counting Sundays**
20. **Factorial digit sum**
21. **Amicable numbers**
22. **Names scores**
23. **Non-abundant sums**
24. **Lexicographic permutations**

**25. N-digit Fibonacci number**

**26. Reciprocal cycles**

**27. Quadratic primes**

**28. Number spiral diagonals**

**29. Distinct powers**

**30. Digit Nth powers**

**31: Coin sums**

**32: Pandigital products**

**33: Digit canceling fractions**

**34. Digit factorials**

**35. Circular primes**

**36. Double-base palindromes**

**37. Truncatable primes**

**38. Pandigital multiples**

**39. Integer right triangles**

**40. Champernowne's constant**

**41. Pandigital prime**

**42. Coded triangle numbers**

**43. Sub-string divisibility**

**44. Pentagon numbers**

**45. Triangular, pentagonal, and hexagonal**

**46. Goldbach's other conjecture**

**47. Distinct primes factors**

**48. Self powers**

**49. Prime permutations**

**50. Consecutive prime sum**

**51. Prime digit replacements**

**52. Permuted multiples**

**53. Combinatoric selections**

**54. Poker hands**

**55. Lychrel numbers**

**56. Powerful digit sum**

**57. Square root convergents**

**58. Spiral primes**

**59. XOR decryption**

**60. Prime pair sets**

**61. Cyclical figurate numbers**

**62. Cubic permutations**

**63. Powerful digit counts**

**64. Odd period square roots**

**65. Convergents of e**

**66. Diophantine equation**

**67. Maximum path sum II**

**68. Magic N-gon ring**

**69. Totient maximum**

**70. Totient permutation**

**71. Ordered fractions**

**72. Counting fractions**

**73. Counting fractions in a range**

**74. Digit factorial chains**

**75. Singular integer right triangles**

**76. Counting summations**

**77. Prime summations**

**78. Coin partitions**

**79. Passcode derivation**

**80. Square root digital expansion**

**81. Path sum: two ways**

**82. Path sum: three ways**

**83. Path sum: four ways**

**84. Monopoly odds**

**85. Counting rectangles**

**86. Cuboid route**

**87. Prime power triples**

**88. Product-sum numbers**

**89. Roman numerals**

**90. Cube digit pairs**

**91. Right triangles with integer coordinates**

**92. Square digit chains**

**93. Arithmetic expressions**

**94. Almost equilateral triangles**

**95. Amicable chains**

**96. Su Doku**

**97. Large non-Mersenne prime**

**98. Anagramic squares**

**99. Largest exponential**

**100. Arranged probability**

**101. Optimum polynomial**

**102. Triangle containment**

**103. Special subset sums: optimum**

**104. Pandigital Fibonacci ends**

**105. Special subset sums: testing**

**106. Special subset sums: meta-testing**

**107. Minimal network**

**108. Diophantine reciprocals I**

**109. Darts**

**110. Diophantine reciprocals II**

**111. Primes with runs**

**112. Bouncy numbers**

**113. Non-bouncy numbers**

**114. Counting block combinations I**

**115. Counting block combinations II**

**116. Red, green or blue tiles**

**117. Red, green, and blue tiles**

**118. Pandigital prime sets**

**119. Digit power sum**

**120. Square remainders**

**121. Disc game prize fund**

**122. Efficient exponentiation**

**123. Prime square remainders**

**124. Ordered radicals**

**125. Palindromic sums**

**126. Cuboid layers**

**127. abc-hits**

**128. Hexagonal tile differences**

**129. Repunit divisibility**

**130. Composites with prime repunit property**

**131. Prime cube partnership**

**132. Large repunit factors**

**133. Repunit nonfactors**

**134. Prime pair connection**

**135. Same differences**

**136. Singleton difference**

**137. Fibonacci golden nuggets**

**138. Special isosceles triangles**

**139. Pythagorean tiles**

**140. Modified Fibonacci golden nuggets**

**141. Investigating progressive numbers, n, which are also square.**

**142. Perfect Square Collection**

**143. Investigating the Torricelli point of a triangle**

**144. Investigating multiple reflections of a laser beam.**

**145. How many reversible numbers are there below one-billion?**

**146. Investigating a Prime Pattern**

**147. Rectangles in cross-hatched grids**

**148. Exploring Pascal's triangle.**

**149. Searching for a maximum-sum subsequence.**

**150. Searching a triangular array for a sub-triangle having minimum-sum.**

**151. Paper sheets of standard sizes: an expected-value problem.**

**152. Writing 1/2 as a sum of inverse squares**

**153. Investigating Gaussian Integers**

**154. Exploring Pascal's pyramid.**

**155. Counting Capacitor Circuits.**

**156. Counting Digits**

**157. Solving the diophantine equation $1/a + 1/b = p/10^n$**

**158. Exploring strings**

**159. Digital root sums of factorisations.**

**160. Factorial trailing digits**

**161. Triominoes**

**162. Hexadecimal numbers**

**163. Cross-hatched triangles**

**164. Numbers for which no three consecutive digits have a sum greater than a given value.**

**165. Intersections**

**166. Criss Cross**

**167. Investigating Ulam sequences**

**168. Number Rotations**

**169. Exploring the number of different ways a number can be expressed as a sum of powers of 2.**

**170. Find the largest 0 to 9 pandigital that can be formed by concatenating products.**

**171. Finding numbers for which the sum of the squares of the digits is a square**

**172. Investigating numbers with few repeated digits**

**173. Using up to one million tiles how many different "hollow" square laminae can be formed?**

**174. Counting the number of "hollow" square laminae that can form one, two, three, ... distinct arrangements.**

**175. Fractions involving the number of different ways a number can be expressed as a sum of powers of 2.**

**176. Rectangular triangles that share a cathetus.**

**177. Integer angled Quadrilaterals.**

**178. Step Numbers**

**179. Consecutive positive divisors**

**180. Rational zeros of a function of three variables.**

**181. Investigating in how many ways objects of two different colours can be**

**182. RSA Encryption**

**183. Maximum product of parts**

**184. Triangles containing the origin.**

**185. Number Mind**

**186. Connectedness of a network.**

**187. Semiprimes**

**188. The hyperexponentiation of a number**

**189. Tri-colouring a triangular grid**

**190. Maximising a weighted product**

**191. Prize Strings**

**192. Best Approximations**

**193. Squarefree Numbers**

**194. Coloured Configurations**

**195. Inscribed circles of triangles with one angle of 60 degrees**

**196. Prime triplets**

**197. Investigating the behaviour of a recursively defined sequence**

**198. Ambiguous Numbers**

**199. Iterative Circle Packing**

**200. Find the 200th prime-proof sqube containing the contiguous sub-string "200"**

**201. Subsets with a unique sum**

**202. Laserbeam**

**203. Squarefree Binomial Coefficients**

**204. Generalised Hamming Numbers**

**205. Dice Game**

**206. Concealed Square**

**207. Integer partition equations**

**208. Robot Walks**

**209. Circular Logic**

**210. Obtuse Angled Triangles**

**211. Divisor Square Sum**

**212. Combined Volume of Cuboids**

**213. Flea Circus**

**214. Totient Chains**

**215. Crack-free Walls**

**216. Investigating the primality of numbers of the form $2n^2 - 1$**

**217. Balanced Numbers**

**218. Perfect right-angled triangles**

**219. Skew-cost coding**

**220. Heighway Dragon**

**221. Alexandrian Integers**

**222. Sphere Packing**

**223. Almost right-angled triangles I**

**224. Almost right-angled triangles II**

**225. Tribonacci non-divisors**

**226. A Scoop of Blancmange**

**227. The Chase**

**228. Minkowski Sums**

**229. Four Representations using Squares**

**230. Fibonacci Words**

**231. The prime factorisation of binomial coefficients**

**232. The Race**

**233. Lattice points on a circle**

**234. Semidivisible numbers**

**235. An Arithmetic Geometric sequence**

**236. Luxury Hampers**

**237. Tours on a 4 x n playing board**

**238. Infinite string tour**

**239. Twenty-two Foolish Primes**

**240. Top Dice**

**241. Perfection Quotients**

**242. Odd Triplets**

**243. Resilience**

**244. Sliders**

**245. Coresilience**

**246. Tangents to an ellipse**

**247. Squares under a hyperbola**

**248. Numbers for which Euler's totient function equals 13!**

**249. Prime Subset Sums**

**250. 250250**

**251. Cardano Triplets**

**252. Convex Holes**

**253. Tidying up**

**254. Sums of Digit Factorials**

# 1. Multiples of 3 and 5

The task is to find sum of all the multiples of 3 and 5 below N.

Multiples of 3 and Multiples of 5 form an AP.

Adding them you will get the sum of multiples, but 15 is repeated once. So, the final equation is

$$S_3 + S_5 - S_{15}$$

and

$$S_n = n * (firstterm + lastterm)/2$$

# 2. Even Fibonacci numbers

The task is to find the sum of all the Fibanocci number below N.

$$F_m = F_{m-1} - F_{m-2}$$

First step is to generate the Fibonacci number and store in an array since we know the Fibonacci number limit can be reached in fewer iterations.

Every 3rd element in the list is even.

Reason: First 2 numbers (0th and 1st) in the series is 0 and 1.

| N | Even/Odd |
|---|---|
| 0th Number | 0 which is **Even** |
| 1st Number | 1 which is Odd |
| 2nd Number | Even + Odd = Odd |
| 3rd Number | Odd + Odd = **Even** |
| 4th Number | Odd + Even = Odd |
| 5th Number | Even + Odd = Odd |
| 6th Number | Odd + Odd = **Even** |

and so on

So, you don't have to check if every number is even by dividing it by 2 and checking the reminder.

Instead, use a loop that iterates over every 3rd element starting from 0th element and add it to a sum variable.

## 3. Largest prime factor

## 4. Largest palindrome product

## 5. Smallest multiple

The task is to find the smallest multiple or (LCM) of all the numbers till N where N ranges till 40.

The root concept behind this is iterative calculation of LCM done easy.

$$LCM(a, b, c, d) = LCM(LCM(LCM(a, b), c), d)$$

and

$$LCM(a, b) = (a * b)/GCD(a, b)$$

## 6. Sum square difference

The task is to find the absolute difference between the sum of the squares of the first natural numbers N and the square of the sum.

If N is 3 then

$$(1 + 2 + 3)^2 - (1^2 + 2^2 + 3^2) = 22$$

is the answer, absolute difference

The first sequence follows the order

$$(\Sigma n)^2$$

The second sequence follows the order

$$\Sigma n^2$$

Store the result of both and print the absolute difference, either that or Solve them to get a final equation

$$Answer = n * (n + 1) * (n - 1) * (3n - 2)/12$$

and print the same

## 7. $10001^{st}$ prime

The task is to print the Nth prime number where N is less than or equal to 10000.

Since we need to do this process for 1000 more times, storing all the first ten thousand times and printing the nth prime would be more efficient time wise.

To store 10000 prime numbers. Find out 10000th prime number, by fact it is 104730.

Sieve of Eratosthenes method of finding out all the prime numbers below N is very much efficient than any other method. It basically takes a number N below which it must find out primes, declares an array of True values of length N.

Then it initializes a counter, whose square should always be less than N.

Till then the counter keeps incrementing, and multiples of that counter is considered, this number is the index of the list whose value is made False since multiples are not primes.

Ex. If 3 is prime, then 6,9,12,15 are not. So those numbers are used as indices whose values are made False.

We get an array of Boolean values which are False at prime indices.

To find out Nth prime number we need to find Nth False value in the list and that's the solution.

## 8. Largest product in a series

## 9. Special Pythagorean triplet

## 10. Summation of primes

## 11. Largest product in a grid

## 12. Highly divisible triangular number

## 13. Large sum

## 14. Longest Collatz sequence

## 15. Lattice paths

The task is to find the number of ways there can be to travel from top left to the bottom right of a given lattice structure, only being able to travel right or down.

Since there are different combinations of routes that we can draw from top left to the bottom right of the lattice. We can identify that it is a problem dealing with the combination.

If it was all possible routes among $MxN$ lattice, then the number of possible routes would be calculated as

$$(m+1)*(n+1)$$

Since it was only bottom or right traversal route, the number of possible routes follow the pattern

$$^{(m+n)}C_m$$

or

$$^{(m+n)}C_n$$

Combination Table

| M and N | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
| 2 | 3 | 6 | 10 | 15 |
| 3 | 4 | 10 | 20 | 35 |
| 4 | 5 | 15 | 35 | 70 |

Final answer needs to be $^{(m+n)}C_n \% (10^9+7)$

Got to make the combination calculations efficient

## 16. Power digit sum

## 17. Number to words

## 18. Maximum path sum I

## 19. Counting Sundays

## 20. Factorial digit sum

## 21. Amicable numbers

## 22. Names scores

## 23. Non-abundant sums

## 24. Lexicographic permutations

The task is to find the Nth lexicographic permutation of the string 'abcdefghijkl'.

Example: if 'abcde' is the string and if we are required to calculate $14^{th}$ lexicographical permutation, i.e., 'adcbe'

This problem requires the knowledge of **factorial number system**.

If we observe closely,

| | | |
|---|---|---|
| abcde | 1! | That is 1st lexicographic permutation |
| abced | 2! | That is 2nd and abdce is 3rd |
| abedc | 3! | That is 6th and acbde is 7th |
| aedcb | 4! | That is 24th and bacde is 25th |
| edcba | 5! | That is 120th and last lexicographic permutation |

The pattern here is that

| | | |
|---|---|---|
| abcde | 1! | The fifth letter changes after 0! = every 1 permutation |
| abced | 2! | The fourth letter changes after 1! = every 1 permutation |
| abedc | 3! | The third letter changes after 2! = every 2 permutations |
| aedcb | 4! | The second letter changes after 3! = every 6 permutations |
| edcba | 5! | The first letter changes after 4! = every 24 permutations |

If we can find the factoradic representation of the number N, that helps us to find the order of Nth permutation of the string.

Factorial Number System uses factorials instead of powers of 10 as in decimal or powers of binary as in binary number systems

First few numbers represented as factoradics

| Number | Factoradic | Expansion |
|---|---|---|
| 0 | 0 | 0 * 0! |
| 1 | 10 | 1 * 1! + 0 * 0! |
| 2 | 100 | 1 * 2! + 0 * 1! + 0 * 0! |
| 3 | 110 | 1 * 2! + 1 * 1! + 0 * 0! |
| 4 | 200 | 2 * 2! + 0 * 1! + 0 * 0! |

And so on... there is always a unique factoradic for each integer

To calculate factoradic of an integer, keep dividing the integer from an incrementing counter until the quotient is 0, keep reverse track of the reminder and that will be your factoradic of that integer.

Example to find the factoradic of 349,

- 349/1 -> 349 quotient with 0 remainder
- 349/2 -> 174 quotient with 1 remainder
- 174/3 -> 58 quotient with 0 remainder
- 58/4 -> 16 quotient with 2 remainder
- 14/5 -> 3 quotient with 4 remainder
- 2/6 -> 0 quotient with 2 remainder

Factoradic of 349(10) => 242010

Similarly factoradic of 14 is 2100

So to find 14th lexicographic permutation of 'abcde', since the string has 5 characters

Prepend '0' to make factoradic a 5 character number 02100

Now loop through the factoradic number, the ith character serves as an index to fetch characters from the string and delete it from the string.

Ex: 0th character is a in abcde, delete a

2nd character is d in bcde, delete d

1st character is c in bce, delete c

0th character is b in be, delete b

0th character is e in e, delete e

Final string will be adcbe, the 14th lexicographic permutation in 'abcde'.

Special care must be taken to calculate factoradic number, as the number gets bigger, say to calculate factoradic of 13! => 1211109876543210, the factoradic number actually should be treated as an array of 13 elements ['12', '11', '10', '9', '8', '7', '6', '5', '4', '3', '2', '1', '0'], so that reversal operations won't hurt the actual elements and also calculating number of elements is easier because

0123456789101112 when reversed actually becomes 2111019876543210 and also contains 16 characters which are undesirable for our calculations, use list in place of strings to avoid this.

## 25. N-digit Fibonacci number

Task is to find the first Fibonacci number whose length is N.

This technique requires the knowledge of memoization, if we can memoize efficiently, this can be solved.

We calculate all the Fibonacci number in an order, check its length, if the length of the current fibo number is greater than previous fibo number then store in a dictionary the length as key and the Fibonacci number's occurrence number as the value.

Ex: If 12th Fibonacci number is the first Fibonacci number to have 3 digits Then fibo_dict[3] = 12.

Python can handle really long integer addition so carry out this process till you have dict[5000] = 23922 or somewhere.

The next steps are to just access the value whose key is user input, can be written in a single line.

**26. Reciprocal cycles**

**27. Quadratic primes**

**28. Number spiral diagonals**

**29. Distinct powers**

**30. Digit Nth powers**

**31: Coin sums**

**32: Pandigital products**

**33: Digit canceling fractions**

**34. Digit factorials**

**35. Circular primes**

**36. Double-base palindromes**

**37. Truncatable primes**

**38. Pandigital multiples**

**39. Integer right triangles**

**40. Champernowne's constant**

**41. Pandigital prime**

**42. Coded triangle numbers**

**43. Sub-string divisibility**

**44. Pentagon numbers**

**45. Triangular, pentagonal, and hexagonal**

**46. Goldbach's other conjecture**

**47. Distinct primes factors**

**48. Self powers**

**49. Prime permutations**