

Math

Input: 10
Output: false
Explanation: Reads 01 from right to left. Therefore it is not a palindrome.

Two tips. ① 110's reverse is 011
which should be 11 should return False
② You don't need to reverse the whole number. For example.
12321.
temp = temp * 10 + num % 10
num = num / 10
You can stop when temp > num.
here temp = 123 and num = 12.
if temp == num or temp / 10 == num
return True

```
M = ["", "M", "MM", "MMM"]
C = ["", "C", "CC", "CCC", "CD", "D", "DC", "DCC", "DCCC", "CM"]
X = ["", "X", "XX", "XXX", "XL", "L", "LX", "LXX", "LXXX", "XC"]
I = ["", "I", "II", "III", "IV", "V", "VI", "VII", "VIII", "IX"]
return M[num//1000] + C[(num%1000)//100] + X[(num%100)//10] + I[num%10];
```

Input: 38
Output: 2
Explanation: The process is like: 3 + 8 = 11, 1 + 1 = 2.
Since 2 has only one digit, return it.

return 0 if num == 0 else (num - 1) % 9 + 1

2 ~~Add Two Numbers (/problems/add-two-numbers)~~

✓ 13 ~~Roman to Integer (/problems/roman-to-integer)~~

9 ~~Palindrome Number (/problems/palindrome-number)~~

✓ 12 ~~Integer to Roman (/problems/integer-to-roman)~~

7 ~~Reverse Integer (/problems/reverse-integer)~~ $[-2^{31}, 2^{31} - 1]$.

535 ~~Encode and Decode TinyURL (/problems/encode-and-decode-tinyurl)~~

66 ~~Plus One (/problems/plus-one)~~

258 **Add Digits (/problems/add-digits)**

149 ~~Max Points on a Line (/problems/max-points-on-a-line)~~

Will do this in hashtable

✓ 67 ~~Add Binary (/problems/add-binary)~~

202 ~~Happy Number (/problems/happy-number)~~

171 ~~Excel Sheet Column Number (/problems/excel-sheet-column-number)~~

50 ~~Pow(x, n) (/problems/powx-n)~~

172 **Factorial Trailing Zeroes (/problems/factorial-trailing-zeroes)**

No other tricks, can skip

5	10	15	20	25
1	2	3	4	5

res = 5 + 1 for 25

8 ~~String to Integer (atoi) (/problems/str~~

233 **Number of Digit One (/problems/numt**

273 **Integer to English Words (/problems/ir**

728 ~~Self Dividing Numbers (/problems/self~~

319 **Bulb Switcher (/problems/bulb-switch**

223 **Rectangle Area (/problems/rectangle-area)**

29 **Divide Two Integers (/problems/divide-two-integers)**

✓ 268 ~~Missing Number (/problems/missing-number)~~

326 **Power of Three (/problems/power-of-three)**

43 ~~Multiply Strings (/problems/multiply-strings)~~

```
1 class Solution:
2     def countDigitOne(self, n):
3         """
4         :type n: int
5         :rtype: int
6         """
7
8         ones, m = 0, 1
9         while m <= n:
10             ones += (n//m + 8) // 10 * m + (n//m % 10 == 1) * (n%m + 1)
11             m *= 10
12         return ones
```

```
public class Solution {
    public boolean isPowerOfThree(int n) {
        // 1162261467 is 3^19, 3^20 is bigger than int
        return (n > 0 && 1162261467 % n == 0);
    }
}
```

✓ 279 [Perfect Squares \(/problems/perfect-squares\)](#)

335 [Self Crossing \(/problems/self-crossing\)](#)

204 [Count Primes \(/problems/count-primes\)](#)

453 [Minimum Moves to Equal Array Elements \(/problems/minimum-moves-to-equal-array-elements\)](#)

69 [Sqrt\(x\) \(/problems/sqrtx\)](#)

65 [Valid Number \(/problems/valid-number\)](#)

224 [Basic Calculator \(/problems/basic-calculator\)](#)

```
if n < 3:
    return 0
primes = [True] * n
primes[0] = primes[1] = False
for i in range(2, int(n ** 0.5) + 1):
    if primes[i]:
        primes[i * i: n: i] = [False] * len(primes[i * i: n: i])
return sum(primes)
```

Write a program to find the n -th ugly number.

Ugly numbers are **positive numbers** whose prime factors only include **2, 3, 5**.

Example:

Input: $n = 10$
Output: 12
Explanation: 1, 2, 3, 4, 5, 6, 8, 9, 10, 12 is the sequence of the first 10 ugly numbers.

Note:

1. 1 is typically treated as an ugly number.
2. n does not exceed 1690.

413 [Arithmetic Slices \(/problems/arithmetic-slices\)](#)

Example 1:

Input: [1,2,3]
Output: 6

Example 2:

Input: [1,2,3,4]
Output: 24

628 [Maximum Product of Three Numbers \(/problems/maximum-product-of-three-numbers\)](#)

$O(1)$ space $O(n)$ time

166 [Fraction to Recurring Decimal \(/problems/fraction-to-recurring-decimal\)](#)

Example 1:

Input: numerator = 1, denominator = 2
Output: "0.5"

Example 2:

Input: numerator = 2, denominator = 1
Output: "2"

Example 3:

Input: numerator = 2, denominator = 3
Output: "0.(6)"

357 [Count Numbers with Unique Digits \(/problems/count-numbers-with-unique-digits\)](#)

Given a **non-negative** integer n , count all numbers with unique digits, x , where $0 \leq x < 10^n$.

Example:

Input: 2
Output: 91
Explanation: The answer should be the total numbers in the range of $0 \leq x < 100$, excluding 11,22,33,44,55,66,77,88,99

264 [Ugly Number II \(/problems/ugly-number-ii\)](#)

343 [Integer Break \(/problems/integer-break\)](#)

Given a positive integer n , break it into the sum of **at least** two positive integers and maximize the product of those integers. Return the maximum product you can get.

✓ 60 [Permutation Sequence \(/problems/permutation-sequence\)](#)

$$|p2.x - p1.x| + |p2.y - p1.y|$$

Example:

Input:

```
1 - 0 - 0 - 0 - 0 - 1
|   |   |   |   |
0 - 0 - 0 - 0 - 0 - 0
|   |   |   |   |
0 - 0 - 1 - 0 - 0 - 0
```

Output: 6

1. "123"
2. "132"
3. "213"
4. "231"
5. "312"
6. "321"

Given n and k , return the k^{th} permutation sequence.

296 [Best Meeting Point \(/problems/best-meeting-point\)](#)

372 [Super Pow \(/problems/super-pow\)](#)

812 [Largest Triangle Area \(/problems/largest-triangle-area\)](#)

400 [Nth Digit \(/problems/nth-digit\)](#)

368 [Largest Divisible Subset \(/problems/largest-divisible-subset\)](#)

313 [Super Ugly Number \(/problems/super-ugly-number\)](#)

537 [Complex Number Multiplication \(/problems/complex-number-multiplication\)](#)

- 396 [Rotate Function \(/problems/rotate-function\)](/problems/rotate-function)
- 672 [Bulb Switcher II \(/problems/bulb-switcher-ii\)](/problems/bulb-switcher-ii)
- 753 [Cracking the Safe \(/problems/cracking-the-safe\)](/problems/cracking-the-safe)
- 883 [Projection Area of 3D Shapes \(/problems/projection-area-of-3d-shapes\)](/problems/projection-area-of-3d-shapes)
- 829 [Consecutive Numbers Sum \(/problems/consecutive-numbers-sum\)](/problems/consecutive-numbers-sum)
- 553 [Optimal Division \(/problems/optimal-division\)](/problems/optimal-division)
- 942 [DI String Match \(/problems/di-string-match\)](/problems/di-string-match)
- 887 [Super Egg Drop \(/problems/super-egg-drop\)](/problems/super-egg-drop)
- 633 [Sum of Square Numbers \(/problems/sum-of-square-numbers\)](/problems/sum-of-square-numbers)
- 367 [Valid Perfect Square \(/problems/valid-perfect-square\)](/problems/valid-perfect-square)
- 507 [Perfect Number \(/problems/perfect-number\)](/problems/perfect-number)
- 877 [Stone Game \(/problems/stone-game\)](/problems/stone-game)
- 908 [Smallest Range I \(/problems/smallest-range-i\)](/problems/smallest-range-i)
- 645 [Set Mismatch \(/problems/set-mismatch\)](/problems/set-mismatch)
- 441 [Arranging Coins \(/problems/arranging-coins\)](/problems/arranging-coins)
- 462 [Minimum Moves to Equal Array Elements II \(/problems/minimum-moves-to-equal-array-elements-ii\)](/problems/minimum-moves-to-equal-array-elements-ii)
- 598 [Range Addition II \(/problems/range-addition-ii\)](/problems/range-addition-ii)
- 670 [Maximum Swap \(/problems/maximum-swap\)](/problems/maximum-swap)
- 858 [Mirror Reflection \(/problems/mirror-reflection\)](/problems/mirror-reflection)
- 365 [Water and Jug Problem \(/problems/water-and-jug-problem\)](/problems/water-and-jug-problem)
- 423 [Reconstruct Original Digits from English \(/problems/reconstruct-original-digits-from-english\)](/problems/reconstruct-original-digits-from-english)
- 836 [Rectangle Overlap \(/problems/rectangle-overlap\)](/problems/rectangle-overlap)
- 885 [Spiral Matrix III \(/problems/spiral-matrix-iii\)](/problems/spiral-matrix-iii)
- 247 [Strobogrammatic Number II \(/problems/strobogrammatic-number-ii\)](/problems/strobogrammatic-number-ii) 🔒
- 248 [Strobogrammatic Number III \(/problems/strobogrammatic-number-iii\)](/problems/strobogrammatic-number-iii) 🔒

- 592 [Fraction Addition and Subtraction \(/problems/fraction-addition-and-subtraction\)](/problems/fraction-addition-and-subtraction)
- 478 [Generate Random Point in a Circle \(/problems/generate-random-point-in-a-circle\)](/problems/generate-random-point-in-a-circle)
- 246 [Strobogrammatic Number \(/problems/strobogrammatic-number\)](/problems/strobogrammatic-number) 🔒
- 593 [Valid Square \(/problems/valid-square\)](/problems/valid-square)
- 517 [Super Washing Machines \(/problems/super-washing-machines\)](/problems/super-washing-machines)
- 891 [Sum of Subsequence Widths \(/problems/sum-of-subsequence-widths\)](/problems/sum-of-subsequence-widths)
- 754 [Reach a Number \(/problems/reach-a-number\)](/problems/reach-a-number)
- 780 [Reaching Points \(/problems/reaching-points\)](/problems/reaching-points)
- 775 [Global and Local Inversions \(/problems/global-and-local-inversions\)](/problems/global-and-local-inversions)
- 640 [Solve the Equation \(/problems/solve-the-equation\)](/problems/solve-the-equation)
- 397 [Integer Replacement \(/problems/integer-replacement\)](/problems/integer-replacement)
- 360 [Sort Transformed Array \(/problems/sort-transformed-array\)](/problems/sort-transformed-array) 🔒
- 523 [Continuous Subarray Sum \(/problems/continuous-subarray-sum\)](/problems/continuous-subarray-sum)
- 781 [Rabbits in Forest \(/problems/rabbits-in-forest\)](/problems/rabbits-in-forest)
- 892 [Surface Area of 3D Shapes \(/problems/surface-area-of-3d-shapes\)](/problems/surface-area-of-3d-shapes)
- 899 [Orderly Queue \(/problems/orderly-queue\)](/problems/orderly-queue)
- 660 [Remove 9 \(/problems/remove-9\)](/problems/remove-9) 🔒
- 868 [Binary Gap \(/problems/binary-gap\)](/problems/binary-gap)
- 805 [Split Array With Same Average \(/problems/split-array-with-same-average\)](/problems/split-array-with-same-average)
- 483 [Smallest Good Base \(/problems/smallest-good-base\)](/problems/smallest-good-base)
- 910 [Smallest Range II \(/problems/smallest-range-ii\)](/problems/smallest-range-ii)
- 869 [Reordered Power of 2 \(/problems/reordered-power-of-2\)](/problems/reordered-power-of-2)
- 914 [X of a Kind in a Deck of Cards \(/problems/x-of-a-kind-in-a-deck-of-cards\)](/problems/x-of-a-kind-in-a-deck-of-cards)
- 794 [Valid Tic-Tac-Toe State \(/problems/valid-tic-tac-toe-state\)](/problems/valid-tic-tac-toe-state)
- 810 [Chalkboard XOR Game \(/problems/chalkboard-xor-game\)](/problems/chalkboard-xor-game)
- 906 [Super Palindromes \(/problems/super-palindromes\)](/problems/super-palindromes)

- 952 Largest Component Size by Common Factor (/problems/largest-component-size-by-common-factor)
- 634 Find the Derangement of An Array (/problems/find-the-derangement-of-an-array) 🔒
- 782 Transform to Chessboard (/problems/transform-to-chessboard)
- 651 4 Keys Keyboard (/problems/4-keys-keyboard) 🔒
- 878 Nth Magical Number (/problems/nth-magical-number)
- 866 Prime Palindrome (/problems/prime-palindrome)
- 800 Similar RGB Color (/problems/similar-rgb-color) 🔒
- 789 Escape The Ghosts (/problems/escape-the-ghosts)
- 469 Convex Polygon (/problems/convex-polygon) 🔒
- 949 Largest Time for Given Digits (/problems/largest-time-for-given-digits)
- 625 Minimum Factorization (/problems/minimum-factorization) 🔒
- 964 Least Operators to Express Number (/problems/least-operators-to-express-number)
- 573 Squirrel Simulation (/problems/squirrel-simulation) 🔒
- 356 Line Reflection (/problems/line-reflection) 🔒
- 927 Three Equal Parts (/problems/three-equal-parts)
- 902 Numbers At Most N Given Digit Set (/problems/numbers-at-most-n-given-digit-set)
- 963 Minimum Area Rectangle II (/problems/minimum-area-rectangle-ii)