Zero-Logic LeetCode Shortcut Sheet (40 curated problems)

A. Always-True / Always-False / Fixed-Output Problems

S.N.	Problem	Shortcut / Observation	One-Liner / Formula	Link
1	**2396. Strictly Palindromic Number**	Impossible for any $n \geq 4$	`return false;`	🔗 Link
2	**877. Stone Game**	First player always wins	`return true;`	🔗 Link
3	**1025. Divisor Game**	Alice wins if `n` even	`return n % 2 == 0;`	🔗 Link
4	**292. Nim Game**	Lose only if `n % 4 == 0`	`return n % 4 != 0;`	🔗 Link
5	**319. Bulb Switcher**	ON bulbs = perfect squares	`return int(sqrt(n));`	🔗 Link
6	**507. Perfect Number**	Only known small values	`return n in {6,28,496,8128,33550336}`	🔗 Link

B. Mathematical / Formula-Based Shortcuts

S.N.	Problem	Shortcut Idea	One-Liner / Formula	Link
7	**258. Add Digits**	Digital-root trick	`(num-1)%9+1 if num else 0`	🔗
8	**268. Missing Number**	Sum formula	`n*(n+1)//2 - sum(nums)`	🔗
9	**172. Factorial Trailing Zeroes**	Count factors of 5	`while n: res+=n//5; n//=5`	🔗
10	**441. Arranging Coins**	Solve $k(k+1)/2 \leq n$	`int((sqrt(8*n+1)-1)//2)`	🔗
11	**1523. Count Odd Numbers in Interval Range**	Odd count formula	`(high+1)//2 - low//2`	🔗
12	**1281. Subtract Product and Sum of Digits**	Digit product − sum	`prod(d)-sum(d)`	🔗
13	**9. Palindrome Number**	Reverse half digits	`x==int(str(x)[::-1])`	🔗
14	**7. Reverse Integer**	String reverse + bounds	`sign=int(x<0); int(str(abs(x))[::-1])`	🔗
15	**263. Ugly Number**	Divide by 2,3,5	Loop divide until not	🔗
16	**69. Sqrt(x)**	Built-in int(sqrt)	`int(x**0.5)`	🔗

C. Bit-Trick Shortcuts

S.N.	Problem	Shortcut Idea	One-Liner / Formula	Link
17	**231. Power of Two**	Single-bit check	`(n>0) and (n&(n-1))==0`	🔗
18	**326. Power of Three**	Highest power divides n	`(n>0) and 3**19 % n==0`	🔗
19	**342. Power of Four**	Power of 2 + even bit	`(n>0)&(n&(n-1))==0&(n&0xAAAAAAAA)==0`	🔗
20	**191. Number of 1 Bits**	Popcount	`bin(n).count('1')`	🔗
21	**476. Number Complement**	Flip within bit-length	`(1<<n.bit_length())-1-n`	🔗
22	**1009. Complement of Base-10 Integer**	Same idea	`(1<<n.bit_length())-1-n`	🔗
23	**136. Single Number**	XOR all elements	`reduce(xor, nums)`	🔗
24	**389. Find the Difference**	XOR s+t	`chr(reduce(xor,map(ord,s+t)))`	🔗

D. String / Built-in Function Shortcuts

S.N.	Problem	Shortcut Idea	One-Liner / Formula	Link
25	**344. Reverse String**	Slice / reverse()	`s[::-1]`	🔗
26	**1108. Defanging an IP Address**	Replace `.` with `[.]`	`address.replace('.', '[.]')`	🔗
27	**709. To Lower Case**	Built-in	`s.lower()`	🔗
28	**58. Length of Last Word**	Split & take last	`len(s.strip().split()[-1])`	🔗
29	**242. Valid Anagram**	Compare sorted	`sorted(s)==sorted(t)`	🔗
30	**383. Ransom Note**	Counter subset	`not (Counter(a)-Counter(b))`	🔗
31	**389. Find the Difference**	XOR chars	see above	🔗
32	**1662. Check If Two String Arrays Are Equivalent**	Join & compare	`"".join(a)==""".join(b)"`	🔗
33	**557. Reverse Words in a String III**	Reverse each word	`' '.join(w[::-1] for w in s.split())`	🔗
34	**771. Jewels and Stones**	Count chars in set	`sum(c in J for c in S)`	🔗

E. Simple Pattern / Direct Computation

S.N.	Problem	Shortcut Idea	One-Liner / Formula	Link
35	**1342. Number of Steps to Reduce to Zero**	Bitcount trick	`steps = n.bit_count() + n.bit_length() - 1`	🔗
36	**1528. Shuffle String**	Use zip+sort	`"".join([c for _,c in sorted(zip(indices,s))])`	🔗
37	**1221. Split a String in Balanced Strings**	Count L/R diff	Increment when balanced	🔗
38	**1768. Merge Strings Alternately**	Zip shorter, then append rest	`"".join(a+b for a,b in zip(word1,word2)) + word1[len(word2):] + word2[len(word1):]`	🔗
39	**1365. Numbers Smaller Than Current**	Sorted index map	`[rank[x] for x in nums]`	🔗
40	**2114. Maximum Number of Words Found in Sentences**	Max(len(split))	`max(len(s.split()) for s in sentences)`	🔗

Total: 40 Problems
Type: Formula-based, mathematical, pattern, or trivial built-in trick
Skill Target: Logical recognition, not algorithmic depth
Average time per problem: 30 sec – 2 min