**check the question and tell me how to approach that problem solve after we can do java code**

**🔢 Number-Based Problems**

1. Check if a number is prime
2. Generate Fibonacci series
3. Check if a number is a palindrome
4. Find factorial of a number
5. Check if a number is an Armstrong number
6. Find GCD of two numbers
7. Find LCM of two numbers
8. Reverse a number
9. Count digits in a number
10. Sum of digits of a number
11. Check if a number is even or odd
12. Check if a number is a perfect number
13. Check if a number is a strong number
14. Check if a number is a Harshad number
15. Convert decimal to binary
16. Convert binary to decimal
17. Convert decimal to octal
18. Convert decimal to hexadecimal
19. Find the nth prime number
20. Print all prime numbers in a range

**🔁 Pattern Printing**

1. Print a pyramid pattern
2. Print an inverted pyramid
3. Print Floyd’s triangle
4. Print Pascal’s triangle
5. Print a diamond pattern
6. Print a hollow square
7. Print a checkerboard pattern
8. Print a right-angled triangle
9. Print a number triangle
10. Print a star hourglass

**🔄 Loop and Series**

1. Print arithmetic progression
2. Print geometric progression
3. Sum of first N natural numbers
4. Sum of squares of first N numbers
5. Sum of cubes of first N numbers
6. Print all even numbers in a range
7. Print all odd numbers in a range
8. Print numbers divisible by 3 and 5
9. Print factorial series
10. Print Fibonacci series using recursion

**🔣 String Manipulation**

1. Reverse a string
2. Check if a string is a palindrome
3. Count vowels and consonants
4. Remove vowels from a string
5. Convert string to uppercase
6. Convert string to lowercase
7. Count words in a string
8. Find duplicate characters
9. Check if two strings are anagrams
10. Replace spaces with hyphens

**📊 Array Problems**

1. Find the largest element in an array
2. Find the smallest element in an array
3. Find the second largest element
4. Reverse an array
5. Sort an array
6. Remove duplicates from an array
7. Merge two arrays
8. Find the frequency of elements
9. Rotate an array
10. Find the sum of array elements

**🔢 Matrix Problems**

1. Add two matrices
2. Multiply two matrices
3. Transpose a matrix
4. Find the diagonal sum
5. Check if a matrix is symmetric
6. Print boundary elements of a matrix
7. Spiral print a matrix
8. Rotate a matrix 90 degrees
9. Find saddle point in a matrix
10. Search an element in a matrix

**🧠 Logic Building**

1. Swap two numbers without a third variable
2. Find the missing number in an array
3. Find the duplicate number in an array
4. Check if a number is a power of 2
5. Count set bits in a number
6. Find the first non-repeating character
7. Implement a basic calculator
8. Convert Roman numerals to integers
9. Convert integers to Roman numerals
10. Generate random numbers

**🧮 Recursion**

1. Factorial using recursion
2. Fibonacci using recursion
3. GCD using recursion
4. Sum of digits using recursion
5. Reverse a number using recursion
6. Power of a number using recursion
7. Binary search using recursion
8. Tower of Hanoi
9. Print all subsets of a string
10. Generate permutations of a string

**🧰 Miscellaneous**

1. Leap year checker
2. Print calendar for a month
3. Count number of words in a sentence
4. Validate email format
5. Generate a random password
6. Find the longest word in a sentence
7. Implement a simple stopwatch
8. Convert temperature (Celsius ↔ Fahrenheit)
9. Find the day of the week for a given date
10. Create a simple number guessing game

Would you like me to generate a PDF or text file with this list for easy reference? Or would you like code examples for any of these?

You said:

give me some more 200 questions.

**Copilot said:**

**Copilot**

Absolutely! Here's a continuation with **200 more programming practice questions**, categorized for clarity and depth. These are great for improving your skills in Java or any other language.

**🧮 Mathematics & Number Theory (1–40)**

1. Find the sum of all prime numbers in a range
2. Find the product of digits of a number
3. Check if a number is a Kaprekar number
4. Check if a number is a Duck number
5. Check if a number is a Spy number
6. Check if a number is a Neon number
7. Check if a number is a Magic number
8. Check if a number is a Trimorphic number
9. Find the HCF of three numbers
10. Find the LCM of three numbers
11. Convert a number to words
12. Convert a number to Roman numerals
13. Find the digital root of a number
14. Find the sum of the first N odd numbers
15. Find the sum of the first N even numbers
16. Count the number of trailing zeros in factorial
17. Find the largest prime factor of a number
18. Find the smallest prime factor of a number
19. Find the sum of all divisors of a number
20. Count the number of divisors of a number
21. Find the nth triangular number
22. Find the nth pentagonal number
23. Find the nth hexagonal number
24. Find the nth Catalan number
25. Find the nth Bell number
26. Find the nth Lucas number
27. Find the nth Mersenne prime
28. Find the nth Palindromic prime
29. Find the nth Happy number
30. Find the nth Ugly number
31. Find the nth Super Ugly number
32. Find the nth Perfect square
33. Find the nth Perfect cube
34. Find the nth Twin prime pair
35. Find the nth Emirp number
36. Find the nth Automorphic number
37. Find the nth Fermat number
38. Find the nth Fibonacci prime
39. Find the nth Tribonacci number
40. Find the nth Tetrahedral number

**🔁 Loops & Series (41–80)**

1. Print the first N palindrome numbers
2. Print the first N Armstrong numbers
3. Print the first N perfect numbers
4. Print the first N prime numbers
5. Print the first N Fibonacci numbers
6. Print the first N square numbers
7. Print the first N cube numbers
8. Print the first N factorials
9. Print the first N Harshad numbers
10. Print the first N Happy numbers
11. Print the first N Ugly numbers
12. Print the first N twin primes
13. Print the first N emirp numbers
14. Print the first N automorphic numbers
15. Print the first N strong numbers
16. Print the first N spy numbers
17. Print the first N neon numbers
18. Print the first N magic numbers
19. Print the first N trimorphic numbers
20. Print the first N Kaprekar numbers

**🔣 String Challenges (81–120)**

1. Check if a string is a pangram
2. Check if a string is an isogram
3. Check if a string is a heterogram
4. Find the longest palindrome in a string
5. Find the shortest word in a string
6. Find the most frequent character
7. Remove duplicate characters
8. Remove all special characters
9. Remove all digits from a string
10. Count uppercase and lowercase letters
11. Toggle case of each character
12. Find all substrings of a string
13. Find all permutations of a string
14. Find all anagrams of a word in a sentence
15. Check if a string is a rotation of another
16. Check if two strings are one edit apart
17. Implement strstr() function
18. Implement strcmp() function
19. Implement string reversal without built-in methods
20. Implement string compression

**📊 Array & List Challenges (121–160)**

1. Find the median of an array
2. Find the mode of an array
3. Find the kth smallest element
4. Find the kth largest element
5. Find the majority element
6. Find the missing number in a sequence
7. Find the duplicate number in a sequence
8. Find the intersection of two arrays
9. Find the union of two arrays
10. Find the common elements in three arrays
11. Find the longest increasing subsequence
12. Find the longest consecutive sequence
13. Find the maximum subarray sum (Kadane’s Algorithm)
14. Find the equilibrium index
15. Find the peak element
16. Find the first repeating element
17. Find the first non-repeating element
18. Find the frequency of each element
19. Rotate array left by k positions
20. Rotate array right by k positions
21. Move all zeros to the end
22. Move all negative numbers to the beginning
23. Segregate even and odd numbers
24. Sort array of 0s, 1s and 2s (Dutch National Flag)
25. Merge two sorted arrays
26. Find the pair with a given sum
27. Find triplets with zero sum
28. Find the smallest missing positive number
29. Find the maximum product subarray
30. Find the minimum product subarray

**Logic & Bit Manipulation (211–240)**

1. Count the number of 1s in binary representation
2. Check if a number is a power of 4
3. Swap two numbers using XOR
4. Find the only non-repeating element in an array
5. Find two non-repeating elements in an array
6. Find the element that appears once when others appear thrice
7. Count the number of bits to convert A to B
8. Check if a number has alternating bits
9. Add two binary numbers
10. Multiply two binary numbers
11. Divide two binary numbers
12. Convert binary to Gray code
13. Convert Gray code to binary
14. Find XOR of all elements in an array
15. Find the missing number using XOR
16. Check if a number is sparse
17. Find the position of the rightmost set bit
18. Turn off the rightmost set bit
19. Turn on the rightmost 0 bit
20. Count total set bits from 1 to N

**🧩 Recursion & Backtracking (241–270)**

1. Solve N-Queens problem
2. Solve Sudoku puzzle
3. Generate all subsets of a set
4. Generate all permutations of an array
5. Generate all combinations of a string
6. Find all paths in a maze
7. Rat in a maze problem
8. Word break problem
9. Subset sum problem
10. Count ways to climb stairs
11. Count ways to reach the end of a board
12. Count ways to tile a floor
13. Count ways to partition a set
14. Count binary strings without consecutive 1s
15. Generate balanced parentheses
16. Find all palindromic partitions of a string
17. Solve Knight’s tour problem
18. Solve Tower of Hanoi with 3+ rods
19. Print all valid IP addresses from a string
20. Print all combinations of coin change

**🧮 Mathematical Algorithms (271–290)**

1. Sieve of Eratosthenes
2. Segmented Sieve
3. Prime factorization using Sieve
4. Modular exponentiation
5. Modular inverse
6. Compute nCr % p
7. Compute nPr
8. Check if a number is a palindrome in any base
9. Count primes less than N
10. Find GCD using Euclidean algorithm
11. Find GCD using Binary GCD
12. Check if a number is a power of another
13. Compute square root using binary search
14. Compute cube root using binary search
15. Compute nth root of a number
16. Count trailing zeros in n!
17. Count digits in n!
18. Compute large factorial using arrays
19. Compute power of a number using recursion
20. Compute power of a number using iteration

**🧰 Data Structures (291–310)**

1. Implement a stack using arrays
2. Implement a stack using linked list
3. Implement a queue using arrays
4. Implement a queue using linked list
5. Implement a circular queue
6. Implement a deque
7. Implement a priority queue
8. Implement a linked list
9. Reverse a linked list
10. Detect loop in a linked list
11. Find the middle of a linked list
12. Merge two sorted linked lists
13. Check if a linked list is a palindrome
14. Remove duplicates from a linked list
15. Find intersection point of two linked lists
16. Implement a binary tree
17. Inorder traversal of binary tree
18. Preorder traversal of binary tree
19. Postorder traversal of binary tree
20. Level order traversal of binary tree
21. Find height of a binary tree
22. Count leaf nodes in a binary tree
23. Check if two trees are identical
24. Convert binary tree to mirror tree
25. Find diameter of a binary tree
26. Find lowest common ancestor
27. Check if a tree is balanced
28. Check if a tree is a BST
29. Find the kth smallest element in BST
30. Serialize and deserialize a binary tree
31. Convert sorted array to BST
32. Convert sorted linked list to BST
33. Print all root-to-leaf paths
34. Find sum of all nodes in a binary tree
35. Find maximum path sum in a binary tree
36. Find vertical order traversal
37. Find top view of a binary tree
38. Find bottom view of a binary tree
39. Find right view of a binary tree
40. Find left view of a binary tree

Would you like me to continue with **questions 311–400**? Or would you prefer a downloadable file (PDF or text) with all 300+ questions so far?

You said:

yes continue

**Copilot said:**

**Copilot**

Awesome! Here's the next set of **100 programming practice questions (311–410)** to continue building your skills.

**🧠 Advanced Data Structures (311–340)**

1. Implement a min-heap
2. Implement a max-heap
3. Heap sort algorithm
4. Find the kth largest element using heap
5. Merge k sorted arrays using heap
6. Find median from data stream
7. Implement a trie (prefix tree)
8. Search a word in a trie
9. Implement autocomplete using trie
10. Implement a hash map
11. Implement a hash set
12. Detect cycle in a graph using DFS
13. Detect cycle in a graph using BFS
14. Topological sort using DFS
15. Topological sort using Kahn’s algorithm
16. Find shortest path using Dijkstra’s algorithm
17. Find shortest path using Bellman-Ford
18. Find shortest path using Floyd-Warshall
19. Find connected components in a graph
20. Check if a graph is bipartite
21. Find articulation points in a graph
22. Find bridges in a graph
23. Implement disjoint set (Union-Find)
24. Find cycle using Union-Find
25. Kruskal’s algorithm for MST
26. Prim’s algorithm for MST
27. Implement LRU cache
28. Implement LFU cache
29. Implement a circular buffer
30. Implement a skip list

**🧮 Math & Geometry (341–370)**

1. Check if three points form a triangle
2. Find area of a triangle using coordinates
3. Find distance between two points
4. Find midpoint of a line segment
5. Check if a point lies inside a triangle
6. Check if a point lies inside a circle
7. Find intersection of two lines
8. Find intersection of two circles
9. Find angle between two vectors
10. Rotate a point around origin
11. Reflect a point across a line
12. Find slope of a line
13. Find equation of a line
14. Find centroid of a triangle
15. Find area of a polygon
16. Check if a polygon is convex
17. Find perimeter of a polygon
18. Find number of diagonals in a polygon
19. Find internal angles of a polygon
20. Convert polar coordinates to Cartesian
21. Convert Cartesian coordinates to polar
22. Find the convex hull of a set of points
23. Implement Bresenham’s line algorithm
24. Implement midpoint circle algorithm
25. Find the bounding box of a shape
26. Find the closest pair of points
27. Find the farthest pair of points
28. Find the diameter of a convex polygon
29. Find the intersection of two rectangles
30. Find the union area of overlapping rectangles

**🧪 Simulation & Games (371–390)**

1. Simulate a dice roll
2. Simulate a coin toss
3. Simulate a deck of cards
4. Implement a number guessing game
5. Implement a rock-paper-scissors game
6. Implement a tic-tac-toe game
7. Implement a simple calculator
8. Simulate a traffic light system
9. Simulate a basic elevator system
10. Simulate a vending machine
11. Simulate a banking system
12. Simulate a parking lot system
13. Simulate a library management system
14. Simulate a hotel booking system
15. Simulate a basic ATM
16. Simulate a basic calendar
17. Simulate a stopwatch
18. Simulate a digital clock
19. Simulate a basic chatbot
20. Simulate a basic file system

**🌐 File Handling & I/O (391–410)**

1. Read a file line by line
2. Count words in a file
3. Count characters in a file
4. Count lines in a file
5. Find the longest word in a file
6. Find the most frequent word in a file
7. Replace a word in a file
8. Remove blank lines from a file
9. Merge two text files
10. Split a file into multiple files
11. Encrypt a file using Caesar cipher
12. Decrypt a file using Caesar cipher
13. Compress a file using RLE
14. Decompress a file using RLE
15. Copy contents from one file to another
16. Append content to a file
17. Read a CSV file
18. Write to a CSV file
19. Read a JSON file
20. Write to a JSON file

**Object-Oriented Programming (411–440)**

1. Create a class for a bank account
2. Implement inheritance with a savings and current account
3. Implement method overloading
4. Implement method overriding
5. Create a class for a student with marks and grade calculation
6. Create a class for a library book with borrow/return methods
7. Create a class for a rectangle with area and perimeter
8. Create a class for a circle with area and circumference
9. Create a class for a car with speed and fuel attributes
10. Create a class for a shopping cart
11. Implement encapsulation in a class
12. Implement abstraction using abstract classes
13. Implement abstraction using interfaces
14. Create a class hierarchy for animals
15. Create a class hierarchy for employees
16. Create a class for a movie ticket booking system
17. Create a class for a hotel reservation system
18. Create a class for a flight booking system
19. Create a class for a food delivery system
20. Create a class for a ride-sharing app
21. Implement a singleton class
22. Implement a factory design pattern
23. Implement a builder design pattern
24. Implement an observer design pattern
25. Implement a strategy design pattern
26. Implement a decorator design pattern
27. Implement a command design pattern
28. Implement a state design pattern
29. Implement a mediator design pattern
30. Implement a prototype design pattern

**🌐 Networking & Web (441–470)**

1. Create a simple HTTP server
2. Create a simple HTTP client
3. Send a GET request to a URL
4. Send a POST request with JSON data
5. Parse JSON response from an API
6. Download a file from the internet
7. Upload a file to a server
8. Create a simple chat server
9. Create a simple chat client
10. Implement a TCP server
11. Implement a TCP client
12. Implement a UDP server
13. Implement a UDP client
14. Create a simple REST API
15. Consume a REST API
16. Parse XML data from a URL
17. Send an email using SMTP
18. Read emails using IMAP
19. Create a simple web scraper
20. Scrape headlines from a news website
21. Scrape product prices from an e-commerce site
22. Scrape weather data from a weather site
23. Scrape job listings from a job portal
24. Scrape stock prices from a finance site
25. Scrape sports scores from a sports site
26. Scrape movie ratings from IMDb
27. Scrape book reviews from Goodreads
28. Scrape restaurant ratings from Zomato
29. Scrape trending topics from Twitter
30. Scrape video titles from YouTube

**🧪 Testing & Debugging (471–490)**

1. Write unit tests for a calculator class
2. Write unit tests for a string utility class
3. Write unit tests for a prime number checker
4. Write unit tests for a palindrome checker
5. Write unit tests for a file reader
6. Write unit tests for a bank account class
7. Write unit tests for a student grading system
8. Write unit tests for a shopping cart
9. Write unit tests for a login system
10. Write unit tests for a REST API
11. Write integration tests for a web app
12. Write functional tests for a login form
13. Write performance tests for a sorting algorithm
14. Write stress tests for a server
15. Write test cases for a calculator
16. Write test cases for a registration form
17. Write test cases for a file upload feature
18. Write test cases for a search feature
19. Write test cases for a payment gateway
20. Write test cases for a booking system

**🧠 Miscellaneous & Fun Projects (491–510)**

1. Create a BMI calculator
2. Create a tip calculator
3. Create a currency converter
4. Create a temperature converter
5. Create a unit converter
6. Create a quiz app
7. Create a to-do list app
8. Create a stopwatch app
9. Create a countdown timer
10. Create a digital clock
11. Create a calendar app
12. Create a note-taking app
13. Create a password generator
14. Create a QR code generator
15. Create a barcode generator
16. Create a weather app using API
17. Create a news app using API
18. Create a dictionary app using API
19. Create a translator app using API
20. Create a stock tracker app using API