**WEEK 2 – LAB ASSIGNMENTS FOR JAVA LAB**

**1. The Buzz of the Century:** You’re a detective tracking down a mysterious code known as a "Buzz number." A Buzz number has a peculiar trait: it's either divisible by 7 or contains the digit 7 somewhere in its sequence. But there's a catch—sometimes, numbers play tricks, and it’s up to you to write a JAVA program to determine if a given number is genuinely a Buzz number. Beware, not all that ends in 7 is truly a Buzz!

**2. The Magic of Factorials:** In the mystical land of Factorialia, you’ve been tasked with calculating the factorial of the number 12. But there's a twist—Factorialia’s rules dictate that your JAVA program must handle numbers much larger than 12 as well. Prepare your code for an epic journey beyond the bounds of simple arithmetic, where numbers grow faster than you can imagine.

**3. Fibonacci’s Golden Sequence:** You’ve stumbled upon an ancient scroll detailing the Fibonacci sequence, but parts of it are missing. The sequence starts with 0 and 1, but to recover the lost numbers, you must write a JAVA program that not only generates the sequence but also identifies specific positions in the sequence where hidden patterns might be found. What secrets will Fibonacci’s Golden Sequence reveal?

**4. The Mirror of Numbers:** In the Enchanted Forest, numbers are said to change form when reflected in the Mirror of Reversal. But not all numbers survive this transformation. Your task is to write a JAVA program that not only reverses a number but checks if the reversed number has any special properties, such as being a prime or a palindrome. Can you uncover what lies beyond the mirror?

**5. The Gatekeeper of Knowledge:** At the gates of a prestigious academy, students are vying for admission to an elite professional course. The gatekeeper has a list of strict criteria: high marks in Mathematics, Physics, and Chemistry are required. But there’s a twist—some students might qualify even if they excel only in Mathematics and Physics. Can you write a JAVA program that processes the applications and identifies the truly exceptional candidates? Only those who meet the rigorous conditions—or perhaps bend them slightly—will pass through.

(a) marks in Mathematics >= 60 (b) marks in Physics >=50

(c) marks in Chemistry >=40 (d) Total in all 3 subjects >=200

(Or)

(e)Total in Maths & Physics>=150

**6. The Quadratic Key:** A riddle has been carved into the ancient walls of the Mathematical Maze: a quadratic equation that must be solved to unlock the way forward. But beware—sometimes the roots are real, other times they are complex, and occasionally, they do not exist at all. Your challenge is to write a JAVA program that not only finds the roots but also interprets what they mean for the path ahead.

**7. The Treasure of Natural Numbers:** Buried deep within the Cavern of Numbers lies a treasure—the sum of natural numbers up to a certain point. But not every treasure map leads to gold. The range is unknown and can change without warning. Write a JAVA program to calculate the sum of natural numbers, but be prepared to handle ranges that stretch far and wide, and perhaps, even down to zero. Will you find the treasure, or will you come up empty-handed?

**8. The Highway of Multiples:** On the Highway of Arithmetic, only multiples of 10 can travel safely between two mystical gates. However, the gates can shift, and the road may span vast distances. Write a JAVA program to identify all multiples of 10 within a given range, but beware—if you miss even one multiple, the journey could end in disaster.

**9. The Table of Multiplication:** In the Kingdom of Multiplication, scholars seek to create tables that guide their learning. But there’s a problem—sometimes, the tables go beyond the ordinary, stretching into dimensions not usually explored. Write a JAVA program that generates a multiplication table for any number, but ensure it can handle unusual requests—like generating tables for very large numbers or printing the table in reverse order. Only the most versatile table will satisfy the scholars.

**10. The Quest for the Greatest Divisor:** Two rival tribes have set out on a quest to find their greatest common factor (GCF), a number that can unite them by dividing both without leaving a remainder. However, the journey is fraught with obstacles—sometimes the numbers are large, and other times, they’re too close to one another. Write a JAVA program that not only calculates the GCF but does so efficiently, even when the numbers challenge the very limits of your code.

**11. The Uniting Multiple:** In a land divided by numbers, two factions wish to unite under a single banner—the smallest multiple of both their chosen numbers. But unity doesn’t come easily; your JAVA program must calculate the lowest common multiple (LCM), even when the numbers seem irreconcilable. Will you find the number that brings harmony, or will the land remain divided?

**12. The Counting Sorcerer:** A powerful sorcerer has summoned you to count the digits of a number, but there’s a trick—some numbers appear simple on the surface but hide their true length behind leading zeros. Write a JAVA program that accurately counts the digits, no matter what form the number takes, and remember—the sorcerer’s test is not as easy as it seems.

**13. The Power of Exponents:** High atop the Mountain of Exponents, a wizard controls the growth of numbers through exponential magic. But the mountain is treacherous, and the wizard's spells can get out of control. Write a JAVA program to calculate the exponential value of any number, but be warned—small mistakes can lead to numbers spiraling out of control, far beyond what you anticipated.

**14. The Palindrome Enigma:** Deep within the Puzzle Palace lies the Palindrome Enigma—a challenge where numbers reflect perfectly or not at all. Your task is to write a JAVA program to determine if a number is a palindrome, but there's a twist—some numbers might be palindromes only after certain transformations, like reversing just part of the number. Can you solve the enigma and find the true palindromes?

**15. The Prime Guardian:** Guarding the gates of the Prime Fortress is a Prime Number, one that can only be divided by 1 and itself. But this guardian is tricky—sometimes it hides among composite numbers, daring you to find it. Write a JAVA program that not only checks if a number is prime but can also detect primes among a list of numbers. Only by revealing the true primes will you earn passage through the fortress.

**16. The Binary Code:** In a digital world where numbers exist in two forms, Binary and Decimal, you are the Keeper of the Code. Your task is to write a JAVA program that can flawlessly convert a Decimal number to Binary and back again. But there's a catch—some numbers are trickier to convert than others, and you must handle edge cases where the conversion isn't straightforward. Can you crack the binary code?

**17. The Median of Harmony:** In the bustling market of Mediania, traders are constantly trying to find the balance—the median price that keeps both buyers and sellers happy. But in this market, the numbers are never evenly spaced, and sometimes, the median isn't as clear as it seems. Write a JAVA program that calculates the median of a set of numbers, even when the set is skewed, large, or contains duplicates. Only then will you bring harmony to the market.

**18. The Infinite e:** The mathematical monks of the Temple of Euler have discovered an infinite series that approximates the number "e," the base of natural logarithms. But the series never truly ends, and as you calculate more terms, the value of "e" becomes more precise. Write a JAVA program to compute "e" using the series expansion, but be prepared to decide when to stop. Will you find the true value of "e," or will you get lost in the infinite?

**19. The Trio Combinations:** In the Game of Numbers, you’ve been tasked with generating all possible combinations of the numbers 1, 2, and 3. But this is no ordinary game—some combinations might be repeated, and others might be missing entirely if you’re not careful. Write a JAVA program that generates all combinations, ensuring that no possibility is overlooked. Only then will you master the game.

**20. The Multiple Mystery:** In the land of Mysterion, two numbers, m and n, have fallen into a dispute. One claims to be a multiple of the other, but the relationship between them isn’t always clear. Sometimes, the multiple is hidden within larger patterns or only emerges after performing calculations. Write a JAVA program to determine if m is a multiple of n, even when the answer isn’t immediately obvious. Solve the mystery, and bring peace to the land.

**21. The Prime Interval:** On a journey through the Desert of Numbers, you come across an interval—a range of numbers that might contain hidden primes. But finding these primes isn’t easy—the desert is vast, and not all numbers within the interval are truly prime. Write a JAVA program to identify all prime numbers within a given interval, but be ready for false leads and tricky calculations. Only the true primes will guide you to your destination.

**22. The Armstrong Challenge:** Legends speak of a number with immense power—a number where the sum of its digits, each raised to the power of the number of digits, equals the number itself. But this power is rare, and many numbers claim to be Armstrong numbers without truly possessing the power. Write a JAVA program to determine if a given number is an Armstrong number, but be vigilant—some numbers might try to deceive you with close approximations. Only by finding the true Armstrong numbers will you complete the challenge.