Basic If–Else Problems:

1. Write a program to check whether a number is positive, negative, or zero.
2. Write a program to check whether a number is even or odd.
3. Write a program to check if a given year is a leap year or not.
4. Write a program to find the greatest of two numbers.
5. Write a program to check whether a person is eligible to vote (age >= 18).
6. Write a program to check whether a given character is a vowel or consonant.
7. Write a program to check if a number is divisible by 5.
8. Write a program to determine whether a given number is a single-digit, two-digit, or more than two-digit number.
9. Write a program to check whether a student has passed or failed (passing marks = 40).
10. Write a program to find whether the entered number is a multiple of both 3 and 7.

Ladder If & Nested If:

1. Write a program to find the greatest among three numbers.
2. Write a program to classify a person based on age: Child (<13), Teenager (13-19), Adult (20-59), Senior (60+).
3. Write a program to assign grades based on marks:
   90-100: A,
   75-89: B,
   50-74: C,
   35-49: D,
   <35: Fail.
4. Write a program to check the type of triangle (equilateral, isosceles, or scalene) based on sides.
5. Write a program to check if a character is uppercase, lowercase, digit, or special symbol.
6. Write a program to calculate electricity bill based on units:
   Up to 100 units: ₹5 per unit,
   101–200 units: ₹7 per unit,
   Above 200 units: ₹10 per unit.
7. Write a program to determine the largest of four numbers using nested if.
8. Write a program to check if a given year is a century year and also a leap year.
9. Write a program to classify BMI value: Underweight (<18.5), Normal (18.5-24.9), Overweight (25-29.9), Obese (30+).
10. Write a program to display the smallest number among three using nested if.

For Loop Problems:

1. Write a program using a **for loop** to print all Armstrong numbers between 100 and 999. (Armstrong number: sum of cubes of digits equals the number itself. Example: 153 => $1^3+5^3+3^3$ = 153).
2. Write a program to generate and display the first **n** prime numbers using a **for loop**.
3. Write a program to display all numbers from 1 to 500 that are divisible by 3, but the sum of their digits should not exceed 10.
4. Write a program using a **for loop** to print a pyramid of stars (*) of height n. Example for n=4:

```
   *
  ***
 *****
*******
```

5. Write a program to accept a string and check whether it is a pangram (contains all 26 alphabets at least once) using a **for loop**.
6. Write a program using a **for loop** to print all twin primes between 1 and 100. (Twin primes: pairs of prime numbers with a difference of 2, e.g., (3,5), (11,13)).
7. Write a program that accepts a number from the user and prints whether it is a Harshad number (number divisible by the sum of its digits) using a **for loop**.
8. Write a program to generate Pascal's Triangle up to n rows using a **for loop**.
9. Write a program using a **for loop** to display the sum of the series:
   $1^2 + 2^2 + 3^2 + ... + n^2$
10. Write a program that accepts a number from the user and prints whether it is a Strong number (sum of factorials of digits = number itself) using a **for loop**. Example: 145 => 1! + 4! + 5! = 145.

<u>While Loop Problems:</u>

11. Write a program using a **while loop** to find the reverse of a number and check if the reversed number is prime. Example: Input = 73 → Reverse = 37 → Prime.
12. Write a program that continues to accept numbers from the user until the sum of digits of all numbers entered becomes greater than 100.
13. Write a program using a **while loop** to check whether a number is a Duck number (a number containing zero but not starting with zero, e.g., 202, 1203).
14. Write a program using a **while loop** to accept a number and check if it is a Happy number. (A number is happy if repeatedly replacing it with the sum of squares of its digits eventually reaches 1). Example: 19 is a happy number.
15. Write a program using a **while loop** to find the largest prime factor of a given number.
16. Write a program to repeatedly accept a string from the user until the string entered is a palindrome.
17. Write a program using a **while loop** to compute the sum of digits of a number until the result becomes a single-digit number (Digital root). Example: 9875 => 9+8+7+5=29 => 2+9=11 => 1+1=2.
18. Write a program using a **while loop** to generate the Collatz sequence for a given number. (Rule: If n is even => n/2, if odd => 3n+1. Continue until n=1).
19. Write a program using a **while loop** to accept a number and check whether it is a Kaprekar number. (Kaprekar number: if square of the number can be split into two parts whose sum equals the number. Example: 45^2=2025 => 20+25=45).
20. Write a program to simulate an ATM machine using a **while loop** where a user can:
    - Check balance
    - Deposit money
    - Withdraw money (only if balance is sufficient)
    - Exit

      Continue until the user chooses to exit.