🔹 1. Find the Largest of Two Numbers

Problem:

Take two numbers and print the larger one.

Algorithm:

1. Start
2. Input two numbers: `num1`, `num2`
3. If `num1 > num2`, then print `num1`
4. Else, print `num2`
5. End

🔹 2. Check if a Number is Even or Odd

Problem:

Check whether a given number is even or odd.

Algorithm:

1. Start
2. Input a number `n`
3. If `n % 2 == 0`, then print "Even"
4. Else, print "Odd"
5. End

🔹 3. Calculate Sum of First N Natural Numbers

Problem:

Find the sum of first `N` natural numbers (like 1 + 2 + 3 + ... + N)

Algorithm:

1. Start
2. Input number `N`
3. Initialize `sum = 0`
4. Repeat from `i = 1` to `N`:
    - Add `i` to `sum`
5. Print `sum`
6. End

🔹 **4. Print Multiplication Table of a Number**

Problem:

Print the multiplication table of a given number up to 10.

Algorithm:

1. Start
2. Input number `n`
3. Repeat from `i = 1` to `10`:
    - Print `n * i`
4. End

🔹 **5. Check if a Number is Positive, Negative or Zero**

Problem:

Determine whether a number is positive, negative, or zero.

Algorithm:

1. Start
2. Input number `n`
3. If `n > 0`, print "Positive"
4. Else if `n < 0`, print "Negative"
5. Else, print "Zero"
6. End

🔹 **6. Find Factorial of a Number**

Problem:

Find the factorial of a number `n` (i.e., `n! = 1 * 2 * 3 * ... * n`)

Algorithm:

1. Start
2. Input number `n`
3. Initialize `fact = 1`
4. Repeat from `i = 1` to `n`:
    - Multiply `fact = fact * i`
5. Print `fact`
6. End

🔹 **7. Check if a Number is Prime**

Problem:

Check whether a number `n` is prime or not.

Algorithm:

1. Start
2. Input number `n`
3. If `n < 2`, print "Not Prime"
4. Repeat from `i = 2` to `n - 1`:
    - If `n % i == 0`, print "Not Prime" and stop
5. If loop completes, print "Prime"
6. End

🔹 **8. Reverse a Number**

Problem:

Reverse the digits of a number.

(E.g. 123 → 321)

Algorithm:

1. Start
2. Input number `n`
3. Initialize `rev = 0`
4. While `n > 0`:
    - `digit = n % 10`
    - `rev = rev * 10 + digit`
    - `n = n // 10`
5. Print `rev`
6. End

🔹 **9. Check for Palindrome Number**

Problem:

Check whether a number is the same when reversed.

(E.g. 121 → palindrome, 123 → not palindrome)

Algorithm:

1. Start
2. Input number `n`
3. Store `original = n`
4. Reverse the number (as in previous example)
5. If `original == reversed`, print "Palindrome"
6. Else, print "Not Palindrome"
7. End

🔹 **10. Find the Smallest Element in an Array**

Problem:

Find the smallest number in a list of elements.

Algorithm:

1. Start
2. Input array/list of numbers
3. Set `min = first element`
4. For each element in array:
    - If element < `min`, set `min = element`
5. Print `min`
6. End

🔹 **11. Count Number of Digits in a Number**

Problem:

Count how many digits are there in a number.

Algorithm:

1. Start
2. Input number `n`
3. Initialize `count = 0`
4. While `n > 0`:
    - `n = n // 10`
    - `count = count + 1`
5. Print `count`
6. End

🔹 **12. Swap Two Numbers (Using Temporary Variable)**

Problem:

Swap the values of two variables.

Algorithm:

1. Start
2. Input `a` and `b`
3. Set `temp = a`
4. Set `a = b`
5. Set `b = temp`
6. Print `a` and `b`
7. End