Q1. Write the algorithm and flowchart of the following:

a. To find the second smallest number from a list of 5 numbers

1. Start
2. Take input a, b, c, d, e
3. If (a < b)
   Then, small = a, ssmall = b
   Otherwise, small = b, ssmall = a
4. If (c < ssmall)
   Then, if (c < small)
      Then, ssmall = small
         small = c
   otherwise, ssmall = c
5. If (d < ssmall)
   Then, if (d < small)
      Then, ssmall = small
         small = d
   otherwise, ssmall = d
6. If (e < ssmall)
   Then, if (e < small)
      Then, ssmall = small
         small = e
   otherwise, ssmall = e
7. Print the value of ssmall
8. Stop

b. To find the lcm and gcd of a number.

LCM:

Algorithm:

1. Start
2. Take input num1 and num2.
3. If (num1 > num2)
   Then, max = num1
   Otherwise, max = num2
4. initialise lcm = max
5. Repeat step 6 and 7 as long as (lcm %num1 != 0 or lcm%num2 != 0) otherwise goto step 8
6. Compute lcm = lcm + max
7. Goto step 5
8. Print the value of lcm as the LCM
9. stop

GCD:

Algorithm:

1. Start
2. Take input num1 and num2
3. Repeat steps 4 to 5 as long as (num2 != 0) otherwise goto step 7
4. initialise temp = num1
5. compute temp = num1
   num1 = num2
   num2 = temp%num2
6. goto step 3
7. print the value of num1 as the GCD
8. stop

## c. To check whether a given number is prime or not.

1. Start
2. Take input num
3. Initialise i = 1, fact = 0
4. Repeat steps 4 to 6 as long as (i < num) other wise go to step 8
5. if (num%i == 0)
   Then, fact = fact +1
6. compute i = i+1
7. goto step 4
8. if (fact ==2)
   Then, print "Prime number"
   otherwise, "Not Prime number"
9. Stop

## d. To display the product of first N natural numbers (use looping).

1. Start
2. Take input num
3. Initialise i = 1, product = 1
4. Repeat steps 4 to 6 as long as (i <= num) otherwise goto step 7
5. Compute product = product * i
   i = i + 1
6. Go to step 4
7. Print the product
8. Stop

## Q2. Explain Von Neumann Architecture.

The von Neumann architecture, proposed by John von Neumann in the mid-20th century, is a foundational design concept for stored-program computers, serving as the basis for most modern computer systems.

1. Central Processing Unit (CPU): The CPU is responsible for executing instructions. It consists of an arithmetic and logic unit (ALU) for performing calculations and logical operations, and a control unit for managing the flow of data and instructions.
2. Memory: The von Neumann architecture uses a single address space to store both data and instructions. This shared memory is used to store the program (instructions) and the data that the program manipulates. This is in contrast to earlier architectures that had separate memory for data and instructions.
3. Input/Output (I/O): External devices, such as keyboards, monitors, and storage devices, are used to input and output data. The CPU communicates with these devices through I/O interfaces.
4. Control Unit: The control unit manages the execution of instructions stored in memory. It fetches instructions from memory, decodes them, and then directs the necessary data and control signals to the ALU and other components to execute the instructions.
5. Bus System: The von Neumann architecture uses a bus system to transfer data and control signals between the CPU, memory, and I/O devices. The bus consists of address lines, data lines, and control lines.

   An important feature is the sequential execution of instructions, where they are fetched and executed one at a time. This design provides a flexible and general-purpose framework for computing, but it can be limited by the sequential nature of instruction execution.