STANDARD 10

COMPUTER SCIENCE

UNIT 01

CHAPTER 01

ALGORITHM FOR PROBLEM SOLVING

1. Define programming.

A: The method of writing the instructions to solve the given problem is called programming.

2. What are the two types of programming techniques?

A: The two types of programming techniques are
(a) Procedural Programming
(b) Object-Oriented Programming

3. What is procedural programming?

A: Procedural programming is a programming technique in which variables are identified and instructions are written using the variables in the correct sequence to get the required result.

4. What is object-oriented programming?

A: Object-oriented programming is a programming technique based on concept of objects which have data related to a person or item.

5. What is an algorithm?

A: The step-by-step instructions required to solve any problem is called an algorithm.

6. Define flowchart.

A: Flowchart is a diagrammatic representation of an algorithm.

7. Explain different symbols used in a flowchart.

A: The different symbols used in the flowchart are

Symbol, Name	Function
 Start, Stop	The ellipse shape indicates the beginning (START) and ending (STOP) of the flowchart.
 Input, Output	The parallelogram shape represents the input and output instructions.
 Processing	A rectangle shape represents the processing step.
 Decision Box	The rhombus shape indicates a point where a decision is to be made.
 Flow of Control	Flow lines with arrow heads indicate the flow of operation, i.e., the exact sequence in which the instructions are to be executed.
 Connector	A connector symbol represents a jump point, indicating that the process continues to another part of the flowchart.

8. What are the advantages of writing algorithms.

A: The advantages of writing algorithms are
(a) Clarity: Provides a clear step-by-step approach to solving problems.
(b) Efficient Problem-Solving: Breaks down complex issues into manageable steps.
(c) Optimization: Helps improve time and space efficiency.
(d) Documentation: Acts as a reference for explaining the solution.
(e) Testing and Debugging: Makes it easier to test and debug code.

9. What are the advantages of drawing flowcharts?

A: The advantages of drawing flowcharts are

(a) Clarity: Flowcharts provide a clear, visual representation of processes.

(b) Simplifies Complex Processes: Breaks down complex processes into simple, manageable steps.

(c) Documentation: Serves as a permanent record of processes for reference, training, and troubleshooting.

(d) Standardizes Procedures: Ensures consistency in performing tasks across teams or departments.

(e) Facilitates Debugging: Makes it easier to trace errors in algorithms or processes.

10. Differentiate between algorithm and flow-chart.

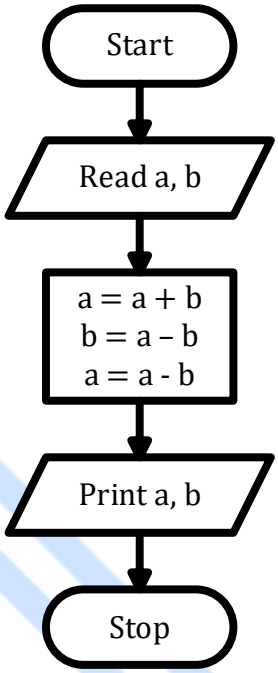
A:

Algorithm	Flowchart
(a)Written in textual form using logical statements.	(a)Diagrammatic, using symbols like rectangles, diamonds, and arrows.
(b)Describes the logic of a solution in written language.	(b)Represents the logic visually through flow symbols.
(c) Requires understanding of programming or logic.	(c)Easier to understand for both technical and non-technical users.
(d) Can be more detailed and may involve complex pseudocode.	(d)Simplifies complex processes into a visual flow of steps.
(e)Used in algorithm design, programming, and problem-solving.	(e)Used in process documentation, decision-making, and programming.

11. Write the algorithm and draw the flowchart to exchange the values of two variables.

A: Algorithm:

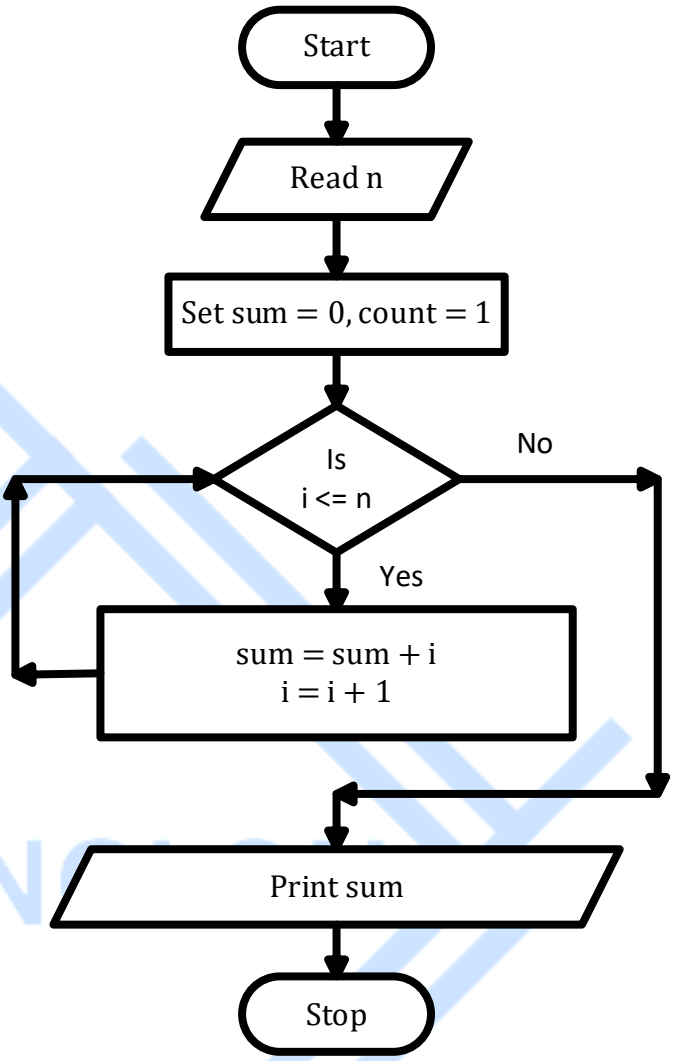
Step 1: Start
Step 2: Input two numbers
A and B
Step 3: Print A and B
Step 4: A = A + B
Step 5: B = A – B
Step 6: A = A – B
Step 7: Print A and B
Step 8: Stop



12. Draw the flowchart and write the algorithm to find the sum of first n natural numbers.

A: Algorithm:

Step 1: Start
Step 2: Input the number N
Step 3: Set i = 1 and Sum = 0
Step 4: Sum = Sum + i
Step 5: i = i + 1
Step 6: If i <= N go to step 4 else go to step 7
Step 7: Print Sum
Step 8: Stop

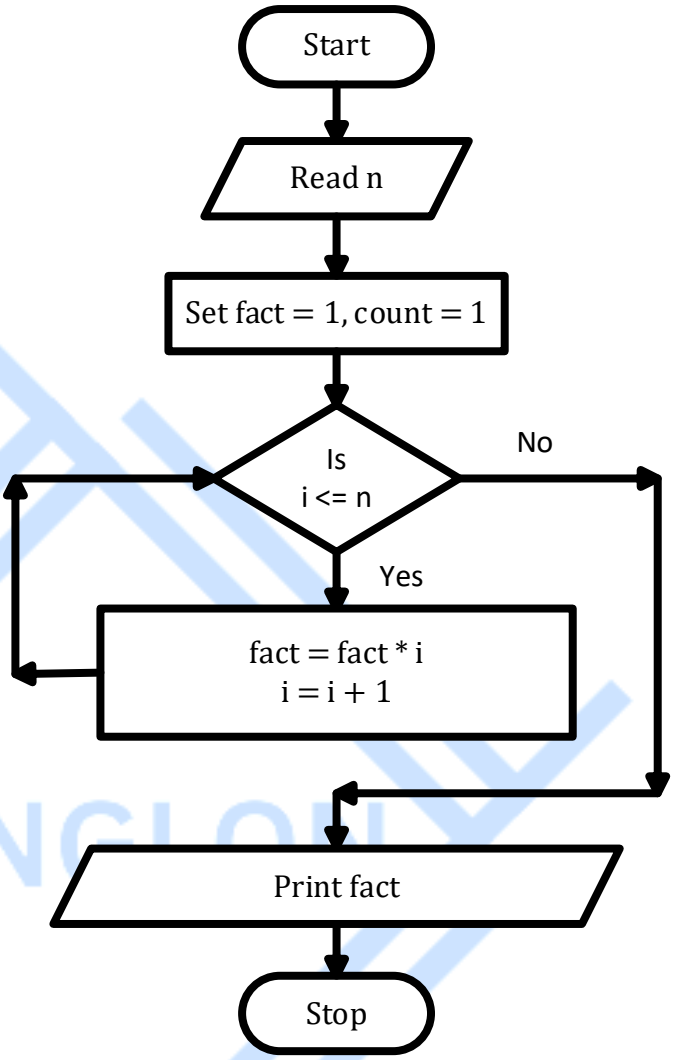


13. Write the algorithm and draw the flowchart to find the factorial of a given number.

A: Algorithm:

Step 1: Start
Step 2: Input the number N
Step 3: Set fact = 1 and count = 1
Step 4: fact = fact * count
Step 5: count = count + 1
Step 6: If count <= N, go to Step 4 else go to Step 7
Step 7: Print fact
Step 8: Stop

Flowchart:

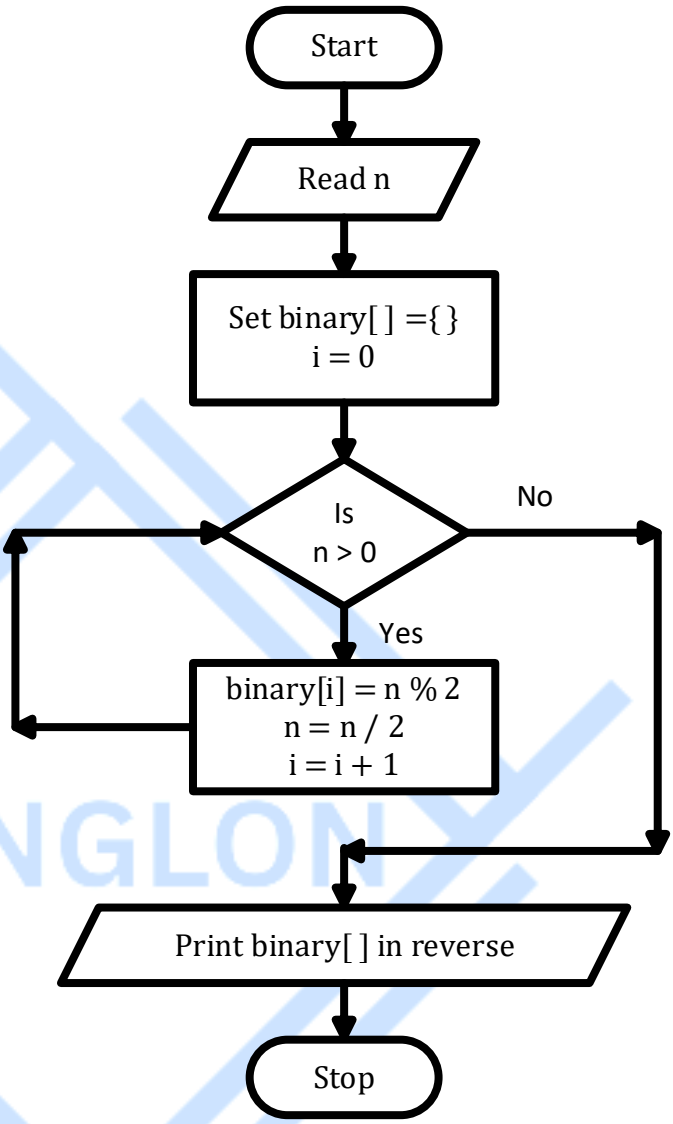


14. Write the algorithm and draw the flowchart to convert decimal number to binary equivalent.

A: Algorithm:

Step 1: Start
Step 2: Input the number N
Step 3: Divide the number by 2 and find the remainder, then store the remainder in an array.
Step 4: Divide the number by 2.
Step 5: Repeat the above two steps until the number is greater than zero.
Step 6: Print the array in reverse order to get the binary representation of the number.
Step 7: Stop

Flowchart:

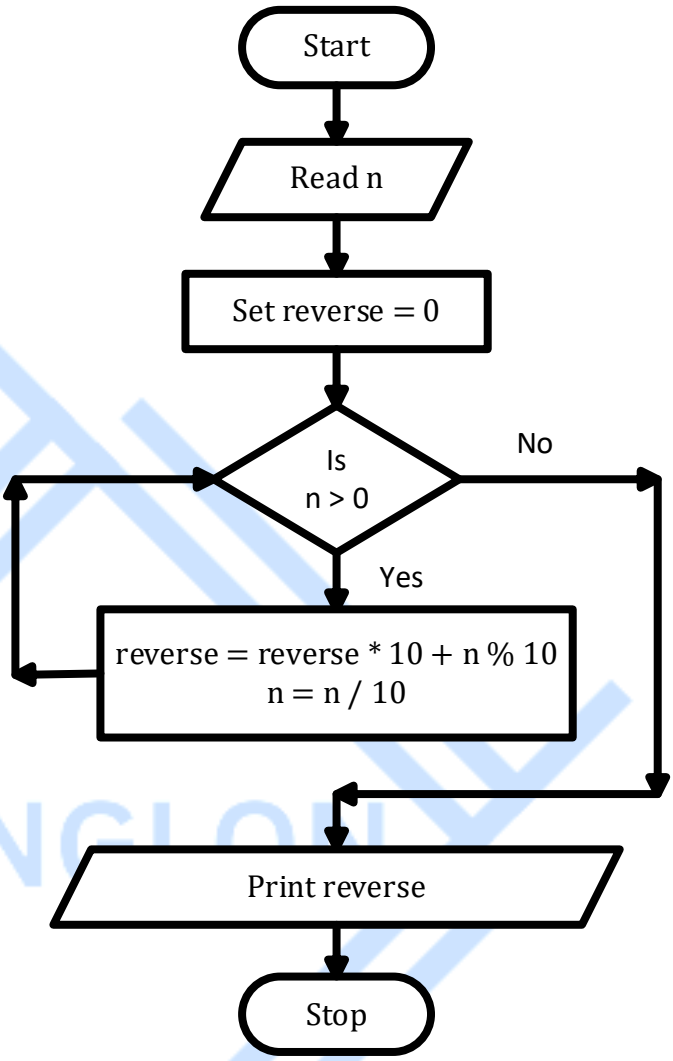


15. Write the algorithm and draw the flowchart to reverse digits of an integer.

A: Algorithm:

Step 1: Start
Step 2: Input the number N
Step 3: Set Reverse = 0
Step 4: If N > 0, go to step 5 else go to step 7
Step 5: Reverse = (Reverse * 10) + (N % 10)
Step 6: N = N / 10 and go to Step 4
Step 7: Print Reverse
Step 8: Stop

Flowchart:

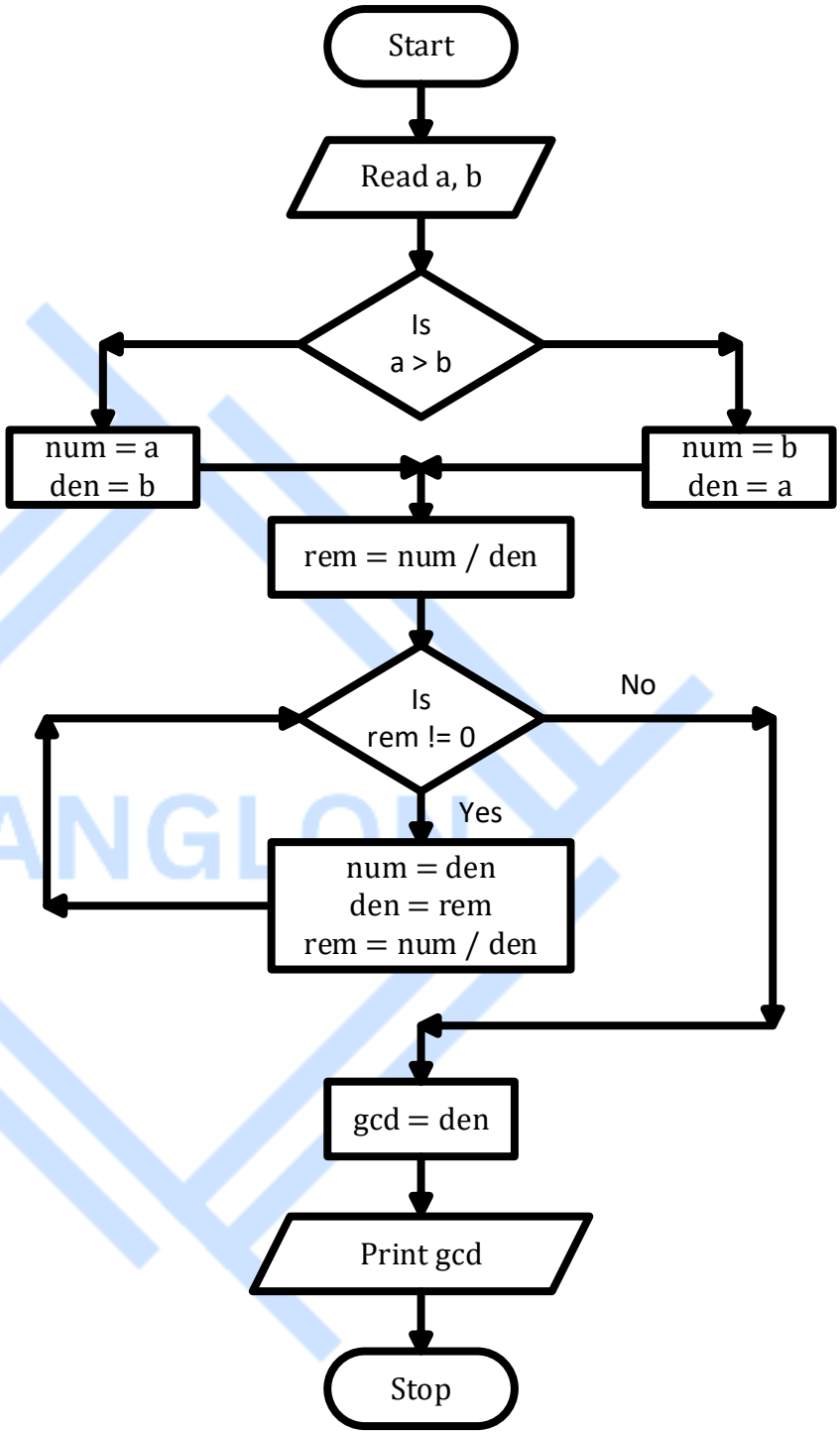


16. Write the algorithm and draw the flowchart to find the GCD/HCF of two integers.

A: Algorithm:

Step 1: Start
Step 2: Input two numbers A and B
Step 3: If A > B, go to Step 5, else go to Step 4
Step 4: Set num = B, den = A and go to Step 6
Step 5: Set num = A and den = B
Step 6: rem = num % den
Step 7: If rem != 0, go to Step 8, else go to Step 10
Step 8: Set num = den and den = rem
Step 9: rem = num % den, go to Step 7
Step 10: Set GCD = den
Step 11: Print GCD
Step 12: Stop

Flowchart:

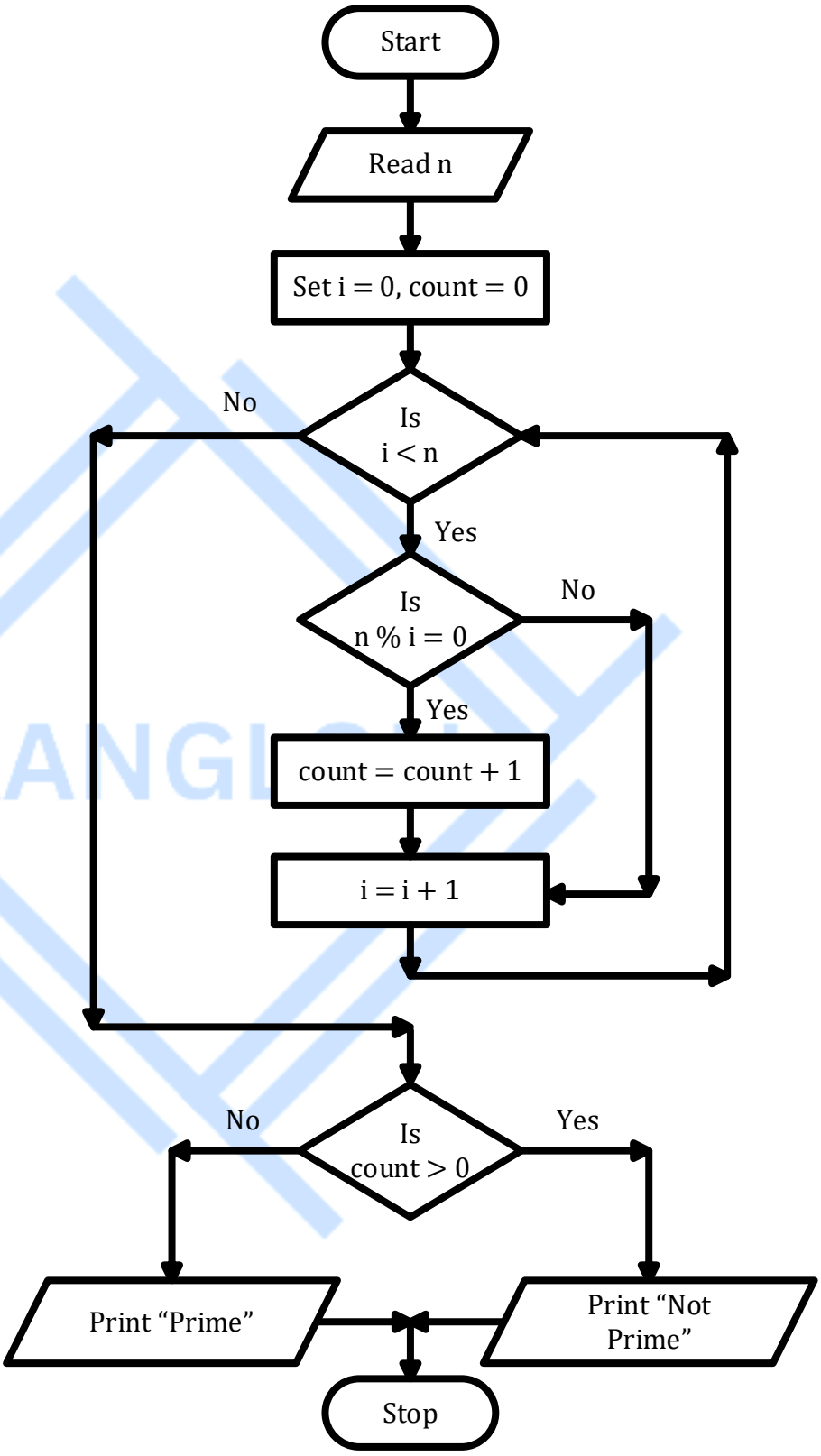


17. Write an algorithm and draw the flowchart to test the primeness of an integer.

A: Algorithm:

Step 1: Start
Step 2: Input the number N
Step 3: Set count = 0 and i = 1
Step 4: If I <= N go to Step 6, else go to Step 8
Step 5: If N % i = 0 go to Step 6, else go to Step 7
Step 6: count = count + 1
Step 7: i = i + 1 and go to Step 4
Step 8: If count = 2, go to Step 9, else go to Step 10
Step 9: Print "Number is prime" and go to Step 11
Step 10: Print "Number is not prime"
Step 11: Stop

Flowchart:

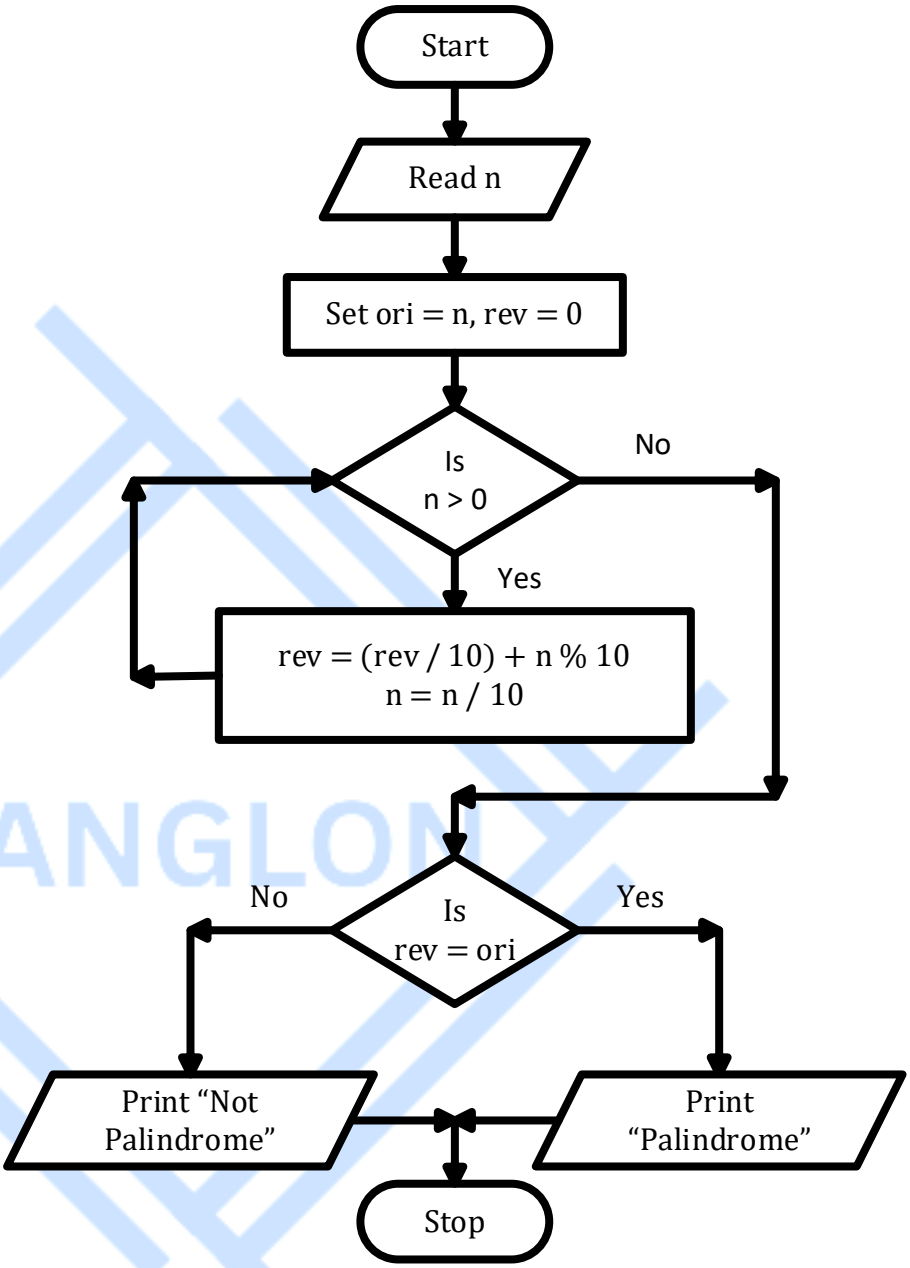


18. Write the algorithm and draw the flowchart to check whether a number is Palindrome or not.

A: Algorithm

Step 1: Start
Step 2: Input the number N
Step 3: original = N
Step 4: Set rev = 0
Step 5: If N > 0, go to Step 6, else go to Step 8
Step 6: rev = (rev * 10) + (N % 10)
Step 7: N = N / 10 and go to Step 5
Step 8: If rev = original, go to Step 9, else go to Step 10
Step 9: Print "The number is palindrome" and go to Step 11
Step 10: Print "The number is not palindrome"
Step 11: Stop

Flowchart:



19. Write an algorithm and draw the flowchart to generate and print the Fibonacci series 0 1 1 2 3 5 8 … n.

A: Algorithm:

Step 1: Start
Step 2: Input the number N
Step 3: Set a = 0, b = 1 and I = 2
Step 4: Print a and b
Step 5: I = I + 1
Step 6: If I <= N, go to Step 7, else go to Step 10
Step 7: next_term = a + b
Step 8: Print next_term
Step 9: Set a = b and b = next_term and go to Step 5
Step 10: Stop

Flowchart: