

# PROGRAMMING

## Section A

1. Define a constant and a variable. Give examples of each.
2. What are the advantages of using library routines?
3. Write pseudocode for a basic IF statement.
4. What does the DIV function do?
5. Explain the difference between COUNT and post-condition loops.
6. How do you declare a variable in Python?
7. What is a case statement, and when would you use it?
8. Give an example of string manipulation using LEFT.
9. What are the three main loop structures in programming?
10. Define “pre-condition” loop with an example.
11. What are the steps to calculate a triangle’s hypotenuse?
12. How do you check if a value is between 10 and 20?
13. Write pseudocode for a basic FOR loop.
14. What is the purpose of using MID in string manipulation?
15. How do you call a procedure in Python?
16. Describe the syntax for defining a function.
17. When should you use a procedure instead of a function?
18. Write pseudocode to check if a number is positive.
19. Explain how parameters work in functions.
20. What is the role of ELSE in an IF statement?
21. Describe a scenario where you’d use a nested IF.
22. What is a menu-driven program?
23. How would you import a library in Java?
24. What does the term “argument” mean in programming?
25. Give an example of a validation check.
26. Why use constants in a program?
27. Explain the purpose of a WHILE loop.
28. Write pseudocode for calculating area and volume of a sphere.

29. What is a default case in a switch statement?
30. Describe the process of debugging.
31. Define “header” in terms of procedures.
32. Write a pseudocode to find the length of a string.
33. Explain the term “library routine.”
34. What is a compiler?
35. How would you end a program when an invalid entry is made?
36. Differentiate between “pass by value” and “pass by reference.”
37. Write pseudocode to calculate simple interest.
38. Explain “return” in functions.
39. Define an infinite loop.
40. What is a constant in Java?
41. Describe how functions improve modularity.
42. When do you use an ELSE IF statement?
43. What is a loop counter?
44. Explain the purpose of declaring variables.
45. Define “initialization” in programming.
46. What are pre-built functions?
47. Write pseudocode to check if an input is an integer.
48. How can a function help in code reusability?
49. Describe how to input data in VB.
50. What is the difference between concatenation and addition in programming?

## Section B

1. Write pseudocode to validate a password based on specific conditions.
2. Explain how a CASE statement can replace nested IFs.
3. How does modular programming help in large projects?
4. What are the benefits of structured programming?
5. Write pseudocode to find the largest of three numbers.
6. Explain the difference between procedure and function headers.
7. Describe an algorithm to sort a list in ascending order.
8. Write pseudocode for a calculator that can add, subtract, multiply, or divide.
9. How would you use a FOR loop for summing integers from 1 to n?
10. What is an identifier table, and why is it useful?
11. Write pseudocode to reverse a string.
12. How does an input-output table help in designing a program?
13. Explain the purpose of an ELSE clause in a loop.
14. What is the purpose of creating flowcharts?
15. Describe the importance of test data.
16. How would you calculate compound interest in pseudocode?
17. Write pseudocode to check if a string is a palindrome.
18. What is a post-condition loop?
19. Explain the function of “break” in loops.
20. Describe how to use a CASE statement in VB.
21. Write pseudocode to count the vowels in a string.
22. What is modularity, and how is it achieved in programming?
23. Define recursion with an example.
24. Write pseudocode for a program that calculates the factorial of a number.
25. Explain the concept of passing arguments by reference.
26. How would you handle errors in input values in a loop?
27. Write pseudocode for a program that outputs Fibonacci numbers.
28. Define abstraction in programming.
29. Write pseudocode to find prime numbers up to 100.
30. Describe error handling in pseudocode.
31. What is a loop invariant?

32. Describe an algorithm to check if a number is even or odd.
33. Explain why nested loops are powerful.
34. Write pseudocode to find the average of numbers.
35. What is scope in terms of variables?
36. Describe the importance of reusability in functions.
37. Explain how to check a string's first and last letters.
38. Write pseudocode to calculate the perimeter of a rectangle.
39. What is the advantage of using library functions?
40. Explain the use of logical operators in IF statements.
41. How does a pre-condition loop work?
42. Write pseudocode for finding the maximum in an array.
43. Describe the role of "return" in functions.
44. What are key differences between Python and Java loops?
45. Define encapsulation and its importance.
46. Write pseudocode for a program that outputs the ASCII values of characters.
47. How is string manipulation handled in VB?
48. What is the purpose of pseudocode?
49. How can functions prevent code redundancy?
50. Describe a situation to use REPEAT ... UNTIL loops.

## Section C

1. Write a pseudocode algorithm for a binary search.
2. Explain the concept of polymorphism with an example.
3. Describe a method to count the occurrences of each character in a string.
4. How do you handle edge cases in algorithms?
5. Write pseudocode to implement bubble sort.
6. Define inheritance and provide a pseudocode example.
7. Explain how recursion works in function calls.
8. What are the time complexities of various sorting algorithms?
9. Describe the concept of encapsulation.
10. Write pseudocode to find the mode of a set of numbers.
11. What is the difference between overloading and overriding?
12. Write pseudocode to implement quicksort.
13. Explain the concept of “divide and conquer” in programming.
14. How does parameter passing work in recursion?
15. Write pseudocode to implement a queue.
16. Describe the algorithm for finding the longest common substring.
17. Write pseudocode to convert infix expressions to postfix.
18. What is big-O notation?
19. Explain the concept of data abstraction.
20. Write pseudocode to simulate a stack.
21. Describe the concept of memoization.
22. Write pseudocode for an algorithm to find the GCD of two numbers.
23. What is tail recursion?
24. Describe the algorithm for Dijkstra’s shortest path.
25. How does garbage collection work in Java?
26. Write pseudocode for finding subsets of a set.
27. Explain dynamic programming with an example.
28. Describe an algorithm for solving the Tower of Hanoi.
29. Write pseudocode for a depth-first search.
30. What is a hash function, and why is it used?
31. Explain the concept of backtracking.
32. Write pseudocode to find the sum of digits in a number recursively.
33. Describe the traveling salesman problem.

34. Explain why sorting is important in searching.
35. Write pseudocode to merge two sorted arrays.
36. What is the significance of “divide and conquer” ?
37. Describe tree traversal methods.
38. Write pseudocode to count all the nodes in a binary tree.
39. How would you implement a hash table?
40. Describe the algorithm for breadth-first search.
41. What are NP-hard problems?
42. Write pseudocode to detect a cycle in a graph.
43. Explain the role of dynamic arrays.
44. Write pseudocode to solve a linear equation.
45. How do binary trees differ from binary search trees?
46. Describe various types of complexity in algorithms.
47. Write pseudocode for matrix multiplication.
48. Explain the significance of modular arithmetic.
49. How does machine learning utilize algorithms?
50. Describe a practical application of recursion.