

Question Bank: Module 3: NumPy, Files, Recursion

Part A – 5 Marks Questions

NumPy

1. Define NumPy. Explain the advantages of NumPy over Python lists.
2. Write a Python program to create a NumPy array and display its shape.
3. Explain the concept of array shape in NumPy with examples.
4. Write a Python program to demonstrate array slicing in NumPy.
5. Explain masking in NumPy arrays with suitable examples.
6. What is broadcasting in NumPy? Explain with examples.
7. Explain the purpose of dtype in NumPy arrays.
8. Write a Python program to create arrays with different data types.
9. Differentiate between Python lists and NumPy arrays.
10. Explain indexing and slicing in NumPy arrays.
11. Write a Python program to perform arithmetic operations using NumPy arrays.
12. Explain the use of Boolean masking in NumPy.
13. Write a short note on multidimensional arrays in NumPy.
14. Explain NumPy array reshaping with examples.
15. Write a Python program to find the sum and average of elements in a NumPy array.

Files

16. Define files in Python. Explain different file modes.
17. Write a Python program to create and write into a text file.
18. Explain reading a file line-by-line with an example.
19. Write a Python program to read the entire contents of a file at once.
20. Explain how to convert a file into a list of lines.
21. Differentiate between text files and binary files.
22. Write a Python program to copy contents from one file to another.
23. Explain working with binary files in Python.
24. Write a short note on directories in Python.
25. Explain the use of os.listdir() with an example.
26. Write a Python program to display files in a directory.
27. Explain fetching data from the Web using Python.
28. Write a Python program to read a file and count the number of words.
29. Explain file handling exceptions in Python.
30. Write a Python program to append data to an existing file.

Recursion

31. Define recursion with an example.
32. Explain recursive data structures.
33. Write a recursive function to find the factorial of a number.
34. Explain recursive processing of number lists.
35. Write a recursive program to calculate Fibonacci numbers.
36. Differentiate between recursion and iteration.
37. Explain base case and recursive case with examples.
38. Write a Python recursive function to find the sum of elements in a list.
39. Explain recursive traversal of directories.
40. Write a recursive program to reverse a string.
41. Explain recursion with stack memory concept.
42. Write a recursive function to find the greatest common divisor (GCD).
43. Explain advantages and disadvantages of recursion.
44. Write a recursive program to count digits in a number.
45. Explain recursive directory handling with examples.

Part B – 10 Marks Questions NumPy

1. Explain NumPy arrays, array creation methods, and advantages with suitable examples.
 - Write a Python program to: create one-dimensional and two-dimensional NumPy arrays,
 - display their shapes and data types.

2. Explain array slicing, indexing, and masking in NumPy with examples.
3. Write a Python program demonstrating broadcasting operations in NumPy.
4. Explain Boolean masking and conditional filtering in NumPy arrays with examples.
5. Write a Python program to perform matrix addition and multiplication using NumPy.
6. Explain the concept of dtype in NumPy with suitable examples.
7. Write a Python program to: reshape arrays - perform slicing - display selected elements.
8. Compare Python lists and NumPy arrays with suitable examples.
9. Explain multidimensional arrays and broadcasting in NumPy with examples.

Files

11. Explain file handling in Python with suitable examples for reading and writing files.
12. Write a Python program to: create a text file, write student details, read and display the contents line-by-line.
13. Explain the following file operations with examples:
 - reading line-by-line,
 - reading the whole file,
 - converting file contents into a list.
14. Write a Python program to count: characters, words, lines in a file.
15. Explain working with binary files in Python with examples.
16. Write a Python program to copy contents from one binary file to another.
17. Explain directory handling in Python using the os module.
18. Write a Python program to recursively display files and folders in a directory.
19. Explain fetching data from the Web using Python libraries with examples.
20. Write a Python program to: open a file, append new data, display the updated contents.
21. Explain different file modes (r, w, a, rb, wb) with examples.
22. Write a Python program demonstrating exception handling in file operations.

Recursion

23. Explain recursion with syntax, flow of execution, and examples.
24. Write a recursive Python program to generate Fibonacci numbers.
25. Explain recursive data structures with suitable examples.
26. Write a recursive program to: find factorial, calculate sum of digits, reverse a string.
27. Explain recursive processing of lists with examples.
28. Write a Python recursive function to search an element in a list.
29. Differentiate between recursion and iteration with suitable examples.
30. Explain recursive traversal of directories and files with examples.
31. Write a Python program to recursively list all files in a directory and its subdirectories.
32. Explain the working of recursive Fibonacci series generation with recursion tree.
33. Write a recursive Python program to calculate power of a number.
34. Explain advantages, disadvantages, and applications of recursion.
35. Write a Python program demonstrating recursive processing of nested number lists.