GE 8151 PROBLEM SOLVING AND PYTHON PROGRAMMING

Important 13mark questions

Unit I

1. Draw a flow chart to accept three distinct numbers, find the greatest and print the result.
2. Write a recursive algorithm to solve towers of Hanoi problem.
3. Mention the different types of iterative structure allowed in Python. Explain the use of continue and break statements with an example.
4. What is an algorithm? Summarise the characteristics of a good algorithm.
5. Outline the algorithm for displaying the first n odd numbers.

Unit II

1. Appraise the arithmetic operators in Python with an example.
2. Sketch the structures of interpreter and compiler. Detail the differences between them. Explain how python works in interactive mode and script mode with examples.
3. Summarize the precedence of mathematical operators in python.
4. Explain the syntax and structure of user defined functions in Python with examples. Also discuss about parameter passing in functions.
5. Describe about the concept of precedence and associativity of operators with example.

Unit III

1. Explain with an example while loop, break statement and continue statement in Python.
2. List the three types of conditional statements and explain them.
3. Write a python code to perform binary search. Trace it with an example of your choice.
4. Analyse string slicing. Illustrate how it is done in Python with example.
5. Outline about function definition and call with example.

Unit IV

1. Appraise the operations for dynamically manipulating dictionaries.
2. Write a Python program to store ‘n’ numbers in a list and sort the list using selection sort.
3. Discuss the different options to traverse a list.
4. Demonstrate with code the various operations that can be performed on tuples.
5. Outline the algorithm and write a Python program to sort the numbers in ascending order using merge sort.

Unit V

1. Tabulate the different modes for opening a file and explain the same.
2. Explain with an example exception with arguments in Python.
3. Discuss about the use of format operator in file processing.
4. Design a Python code to count the number of words in a python file.

5. Explain the commands used to read and write into a file with examples.