[6359]-515

S.E. (Electronics/E&TC/Electronics(VLSI Design & Tech.)/Electronics & Communication-Advanced Communication Technology) (Insem)
DATA STRUCTURES
(2019 Pattern) (Semester-III) (204184)

Time : 1 Hour]   *[Max. Marks : 30*

Instructions to the candidates:

1) *Answer Q1 or Q2, Q3 or Q4.*
2) *Neat diagrams must be drawn wherever necessary.*
3) *Figures to the right indicates full marks.*
4) *Assume suitable data, if necessary.*

Q1) a) Define Pointer.Explain pointer declaration, initialization, pointer arithmetic with suitable example? **[5]**

b) Explain call by value and call by Address for the example of swapping two numbers. **[6]**

c) Explain the difference between structure and union with suitable Example? **[4]**

OR

Q2) a) What is String? How to declare string in C? Write user defined function to calculate length of string? **[5]**

b) List different types of operators in C? Explain Bitwise operator in detail? **[5]**

c) Explain need of File Handling in C? Explain the difference between text and binary file? What are different file handling modes? **[5]**

Q3) a) Write step by step procedure for performing Binary search on following array [10, 22, 35, 40, 45, 50, 80, 82, 85, 90, 100] to search for the element 45. **[5]**

b) Explain Time and Space Complexity? Explain the significance of Big O, Big Theta, and Big Omega notations? **[5]**

c) Compare Bubble, Insertion, Selection, Merge and Quick Sort with respect to stability, worst case time complexity, Adaptivity? **[5]**

OR

Q4) a) Show All steps for performing insertion sort on [12, 15, 17, 11, 9, 13, 18, 16]. **[5]**

b) What will be big(O) for the following code? Write main function to perform addition of n elements entered by user using array for the following code? **[5]**

```
int sum(int arr[], int n)
{
        int i, total = 0;
        for (i =0; i < n; i++) {
        total += arr[i];
        }
        return total;
}
```

c) Compare Linear and Binary Search. Write an Algorithm to search the element in a list using Linear Search. **[5]**

❖❖❖❖❖