2012

MCA

1st Semester Examination

INTRODUCTION TO PROGRAMMING & DATA STRUCTURE

PAPER—MCA-107

(PRACTICAL)

Full Marks : 50

Time : 6 Hours

The figures in the margin indicate full marks.

Candidates are required to give their answers in their own words as far as practicable.

Illustrate the answers wherever necessary.

Introduction to Programming (Group—A)

Answer any *one* question. (Lottery Basis) 1×35

1. Write a program to find print out all the Armstrong numbers between 1 to 1000.

2. Write a program to receive an integer and find it's octal equivalent.

3. A 5-digit number is entered through the keyboard. Write a program to calculate the sum of digits of the 5-digit number using recursion.

4. Write a C program to evaluate the series :

$$\text{Sin}(x) = x - \left(\frac{x^3}{3!}\right) + \left(\frac{x^5}{5!}\right) - \left(\frac{x^7}{7!}\right).$$

5. A 3×3 matrix is entered through keyboard. Write a program to obtain the Determinant value of this matrix.

6. Write a program to delete all vowels from a sentence. Assume that the sentence is not more than 100 characters long.

7. Write a program that compares two given dates. To store a date, use a structure that contains three members namely date, month and year. If the dates are equal, then display message as "Equal" otherwise "Unequal".

8. Write a program to read a file and display its contents along with line numbers before each line.

9. Write a program that takes the contents of a file and copies them into another file, character by character.

10. Write a program that takes the name of an individual and abbreviates the first, middle (if any) and last names by their first letters.

[Viva + PNB = 10 + 05 marks]

Data Structure (Group—B)

Answer any *one* question. (Lottery Basis) 1×35

1. Write a program in C to find a particular element from an array using binary search technique.

2. Write a program in C to sort a number of integers using quick sort algorithm.

3. Write a program in C to sort a number of integers using selection sort algorithm.

4. Create a circular queue that include the following functions : insert and delete.

5. Implement a stack in which push and pop operations may be performed. Demonstrate using ten elements.

6. Implement a binary search tree.

7. Write a program in C to convert an infix expression into it's equivalent postfix expression.

8. Implement bubble sort.

9. Write a program to perform insertion & deletion operations in B-tree.

10. Create a singly linked list and write functions to perform the following operations :

(a) Insert a element before an existing element ;

(b) Count total number of elements in the list ;

(c) Show the elements from the end.

[Viva + PNB = 10 + 05 marks]