Department of Computer Science
Subject: Prog. Fundamentals and C Lang. (MCA C-101)

Time: 3H   MM: 70

Section-A
Attempt any FIVE questions out of TEN. Each question carries 6 marks.

Q1. What is the working principle of prefix operators? Explain with example.   6
Q2. What is Dynamic memory allocation? Explain with example.   6
Q3. What are the various Input/output Statement in C? Explain with example.   6
Q4. List of various Operators in C. Explain with example.   6
Q5. Difference between Call by value and Call by reference. Explain with example.   6
Q6. How are pointer declared and initialized? Explain with example.   6
Q7. Different between if-else and Nested-if statement. Explain with example.   6
Q8. Explain how to display and set the terminal characteristics of a UNIX OS.   6
Q9. Explain the following commands with example:
    i) HOME   ii) cd   iii) pwd   iv) mkdir   v) rmdir   vi) wc   6
Q10. What is the significance of gets and puts functions? Explain with example.   6

Section-B
Attempt any FOUR questions out of EIGHT. Each question carries 10 marks.

Q1. Write a program in C to sort in ascending order integer elements of a one-dimensional array? Explain with example.   10

Q2. How does file support to maintain data in C programming? Write a program to create a file and display its contents. Assume suitable data.   10

Q3. What do you mean by Recursion? What is the Use of recursion in c Programs? Write a program to find factorial of a Number using Recursion.   10

Q4. Describe the following:
    a) Unions and Structures
    b) Macros in C   10

Q5.   A) Describe the use of Arrays in C programs with an example
       B) List of various operators used in C along with its precedence?   10

Q6.   A) What are Function? How are function declared and Parameters passed of Function?
       B) Write a C program to print the average of all numbers from 1 to 50?   10

Q7.   A) Describe how can we store a negative integer?
       B) Differentiate between calloc() and malloc()? Explain with example.   10

Q8. Implement the following procedure to generate Prime numbers from 1 to 100 into a program. This procedure is called sieve of Eratosthenes.
       Step1: Fill an array **num[100]** with numbers from 1 to 100
       Step2: Starting with the second entry in the array, set all its multiples to zero.
       Step3: Proceed to the next non-zero element and set all its multiples to zero.   10