M.C.A. DEGREE I SEMESTER EXAMINATION, DECEMBER 2006

CAS 2103 DATA STRUCTURES WITH C

Time: 3 Hours Maximum marks : 50

PART - A

(Answer ***ALL*** questions)
(Each question carries ***TWO*** marks)

(15 x 2 = 30)

I. (a) What is the use of the keyword "extern" with respect to C ?
(b) What are Escape sequences in C ?
(c) Write a program in C to find the largest and second largest number in a given series of numbers without using an array or any sorting algorithm.

II. (a) Convert the expression from infix to postfix.
Infix String : a+b*c-d
(b) Write an algorithm to implement a queue using arrays ?
(c) What is dequeue ?

III. (a) What is a dangling pointer ?
(b) What do you mean by 'memory leak' ?
(c) What is a doubly linked list ?

IV. (a) Differentiate between a B-tree and a B+-tree.
(b) Differentiate between a binary tree and a threaded binary tree.
(c) Differentiate between a compact and a non compact tree.

V. (a) What is reverse polish notation ?
(b) How is a postfix expression processed ?
(c) Define expression tree ?

PART – B

(Answer ***ALL*** questions)
(Each question carries ***FOUR*** marks)

(5 x 4 = 20)

VI. A. Explain the different control statements in C.

OR

B. Write a complete program in C to count the number of words, lines and characters in a given paragraph.

VII. A. Write a C program to sort a list of strings using an array of pointers whose addresses have to be used for sorting.

OR

B. Write a C program to implement a stack using arrays.

VIII. A. Write a C program to implement a queue using linked list.

OR

B. Write a program in C to sort the numbers accepted from the keyboard using linked list.

IX. A. Describe the (i) inorder (ii) preorder and (iii) postorder traversal of a binary tree using an example.

OR

B. Explain the basic operation of a **B-tree**. Explain the construction of a B-tree of order 3 when the values are inserted in the order (8,5,1,7,3,12,9,6).

X. A. (i) Explain the expression tree.
(ii) Write a function for infix expression tree traversal.

OR

B. Explain the evaluation of a reverse polish expression.

***