

**B.TECH SEM - III (2007 COURSE) (COMPUTER ENGG.) : WINTER
- 2017**

SUBJECT: ALGORITHMS AND DATA STRUCTURES

Day: **Monday**
Date: **15/01/2018**

W-2017-2363

Time: **10.00 AM TO 01.00 PM**
Max. Marks: 80

N.B.:

- 1) **Q. 1 and Q.5** are **COMPULSORY**. Out of the remaining attempt any **TWO** questions from each Section.
- 2) Figures to the right indicate **FULL** Marks.
- 3) Answer to both the sections should be written in **SEPARATE** answer books.
- 4) Draw neat and labeled diagrams **WHEREVER** necessary.

SECTION - I

- Q.1**
- a) What is ADT and Realization of ADT in C. (05)
 - b) Write an algorithm to convert infix expression into postfix expression. (05)
 - c) What is linked list? What are it's advantages over an array? (04)
- Q.2**
- a) What do you mean by ADT? Write an formal definition of ADT for integer. (07)
 - b) Write a notes on (06)
 - i) Concept of class
 - ii) Data definition
 - iii) Abstraction
- Q.3**
- a) Write an algorithm for reversing a given string using stack as data structure. Assume that the necessary stack operations are available manage stack as a linked list. (07)
 - b) Evaluate the following prefix expression using stack for A= 16, B =2, C =3, D = 10 and E = 4. Show status of stack in each step (06)
Prefix expression = - + / A ^ BC * AC.
- Q.4**
- a) Explain the classification of data structures with an example each. (07)
 - b) Write an algorithm to represent sparse matrix using linked list also given on example. (06)

SECTION - I I

- Q.5**
- a) Explain in detail single linked list with example. (05)
 - b) How time complexities arte represented graphically? (05)
 - c) Write pseudo code for merge sort. (04)
- Q.6**
- a) Explain in detail deletion of node in circular linked list? (07)
 - b) How polynomials are manipulated using linked list? (06)
- Q.7**
- a) Why searching is important? Explain any one searching technique in detail. (07)
 - b) How searching is carried out on ordered table? (06)
- Q.8**
- a) Explain shell sort in detail (07)
 - b) Explain quick sort in detail (06)

* * * * *