S.E. (Computer Engg.) (I Semester) EXAMINATION, 2011

DATA STRUCTURES AND ALGORITHMS
(2008 PATTERN)

Time : Three Hours
Maximum Marks : 100

N.B. :—
(i) Answer 3 questions from Section I and 3 questions from Section II.
(ii) Answers to the two sections should be written in separate answer-books.
(iii) Neat diagrams must be drawn wherever necessary.
(iv) Figures to the right indicate full marks.
(v) Assume suitable data, if necessary.

SECTION I

1. (a) Explain the following concepts in ‘C’ language, with example:
   (i) Pointer
   (ii) Call by value
   (iii) Array of pointers
   (iv) Structure.

   [8]
(b) Write a 'C' program to perform the following operations on an array of integers of size 'n', using functions with parameters:

(i) Delete an element at a given position  
(ii) Count and display the prime numbers. [8]

Or

2. (a) Write the output of the following code:

```c
#include <stdio.h>

#define N 5

int main( )
{

    int i,j,n=N;
    int *p, list[ ] = {10, 20, 30, 40, 50};
    char name[][4] = {"ABC", "BCD", "CDE"};
    char(*q)[4];
    p = &list[0];
    q = &name[0];
    for (i=0;i<n;)
        printf ("%d\t%d\n",*(list+i++),*(p++));
    for (j=0;j<3;j++)
        printf("%s\t%c\t%d\n",name[j],*(q+j),name[j][0]);
}
```
[8]
(b) Write a ‘C’ program to read and store data about ‘n’ machine parts in a file. The data of each machine part consists of a part number, name and quantity. Write a ‘C’ function to display the total quantity of all parts and the part numbers having quantity less than a given value. [8]

3. (a) Explain the following terms with an example:
   (i) Static Data structure
   (ii) Omega (Ω) notation
   \[ \text{asymptotically lower bound} \]
   \[ f(x) \geq c \cdot g(x) \text{ whenever } x \geq k \]
   (iii) Flow chart
   (iv) Data object.
   (b) Write an algorithm to find the sum of odd and even integers in an array of integers of size ‘n’. Comment on the data structures used in your algorithm. [8]

4. (a) Determine the efficiency of an algorithm for finding the minimum number from a list using frequency count. [8]
   (b) Give the classification of data structures with respect to memory allocation and relation between elements. [8]

5. (a) Write a pseudo 'C' code to multiply two polynomials using arrays. [6]
(b) Explain the representation of multidimensional arrays in memory. [6]

(c) Write a pseudo ‘C’ code to find the simple transpose of a sparse matrix. [6]

Or

6. (a) What is a sparse matrix? Give the representation of a sparse matrix using arrays. [6]

(b) Write an algorithm to find the multiplication of a polynomial by an integer constant. [6]

(c) Explain the representation of polynomial using array with example. [6]

SECTION II

7. (a) Write a pseudo ‘C’ code for sequential search and analyse the time and space complexity. [8]

(b) Explain the concept of radix sort with pseudocode. [8]

Or

8. (a) Write the pseudocode for binary search. Analyse the time complexity and compare it with sequential search. [8]

(b) Write short notes on:

(i) Selection sort

(ii) Efficiency of sorting. [8]
9. (a) Differentiate between array and linked list. [4]

(b) What is a circular linked list? Write a 'C' function to create a polynomial using circular linked list. [6]

(c) Two lists of numbers are stored in singly linked lists 'A' and 'B'. Write 'C' functions to:
   (i) Concatenate two lists 'A' and 'B'
   (ii) Count the number of nodes in list 'A'
   (iii) Find the sum of numbers in list 'B'. [8]

Or

10. (a) Explain the structure of doubly linked lists. [4]

(b) Write a 'C' function to evaluate a polynomial represented as circular linked list. [8]

(c) Write 'C' functions to store a list of names using linked list and display the list after sorting. [6]

11. (a) Give the implementation of ADT stack using linked list. [8]

(b) What is a double ended queue? Write pseudo 'C' code for operations on dq. [8]
12. (a) Compare linear queue and circular queue. Write the pseudocode for operations on circular queue.

(b) Convert the following infix expression into postfix form and write pseudo 'C' code to evaluate a postfix expression.

\[(A \times B) / C - (B + D - E)\]