

[3862]-214

S.E. (Computer Engineering) (First Semester) EXAMINATION, 2010

DATA STRUCTURES AND ALGORITHMS

(2008 COURSE)

Time : Three Hours

Maximum Marks : 100

N.B. :— (i) Answer any *three* questions from each Section.

(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) What is call by value and call by reference ? Explain with *one* example for each. [8]

(b) Write a 'C' program to shift elements of an integer array one location ahead. Element present at last location of array should be shifted on first location of the array. Write separate functions for accepting array, shifting array, displaying array etc. [8]

Or

2. (a) Explain execution of the following code step by step and show the final output : [8]

```
# include <stdio.h>
```

```
void display (int);
```

```
int main( )
```

```
{   int x = 4;
```

```
    display (x);
```

```
    return 0;
```

```
}
```

```
void display (int n)
```

```
{   if (n > 0)
```

```
    display (n - 1);
```

```
    printf ("%d", n);
```

```
}
```

- (b) Write a 'C' program to implement a structure for automobile part with data members part number, part name, quantity etc. Write separate functions for initializing structure, updating structure, displaying structure etc. [8]
3. (a) Explain in detail the concept of data object, data structures, linear data structure and non-linear data structure. [8]
- (b) What is Asymptotic notation ? Write an algorithm for matrix multiplication for $n \times n$ matrix and find out its time complexity by frequency count. [8]

Or

4. (a) What is Abstract Data Type (ADT) ? Write an abstract data type for linked list. [8]
- (b) Write an algorithm for Bubble sort. Find time complexity of the algorithm and express it using asymptotic notation. [8]
5. (a) (i) Show how a two-dimensional array is stored in memory. Assume that array start at the address 4000. [4]
- (ii) Explain how a polynomial is represented using array with one example. [5]
- (b) Write an algorithm for fast transpose of sparse matrix and find out its time complexity. [9]

Or

6. (a) Write a 'C' program to implement polynomial using array and perform its multiplication. (Write separate functions for accepting polynomial, multiplication, display). [10]
- (b) (i) What is sparse matrix ? Explain with one example. [4]
- (ii) What is address of element $\text{arr}[4][3]$ in the array 'arr' of size 6×6 and type integer, when 'arr' is represented using row major and column major representations ? Assume array 'arr' starts at the address 7000. [4]

SECTION II

7. (a) Sort the following numbers step by step by using Radix sort : [8]
- 20, 15, 21, 06, 08, 05, 29, 02, 14, 40.
- (b) Write an algorithm for binary search and find out its time complexity. [8]

Or

8. (a) Explain with *one* example index sequential search. [8]
(b) Write a 'C' program to implement quicksort. [8]
9. (a) (i) What is static memory allocation and dynamic memory allocation ? [4]
(ii) Compare doubly linked list and circular linked list. [4]
(b) Write an algorithm to delete and insert a node in doubly linked list at any position. [8]

Or

10. (a) (i) What is skip list ? Explain with *one* example. [4]
(ii) Draw GLL for the following expression : [4]
(a, b, (c, d, (e, f), g, (h, i), ((j, k)), l), m)
(b) Write a 'C' program to implement circular linked list and display the contents in reverse order. (Write separate functions for create and display) [8]
11. (a) (i) Define stack and write abstract data type for stack.. [5]
(ii) What is multistack ? Explain with *one* example. [4]
(b) Write a 'C' program to implement circular queue using array and perform insert and delete operation. [9]

Or

12. (a) (i) Differentiate between linear and circular queue. [2]
(ii) Explain stack overflow and underflow conditions. [4]
(iii) What is priority queue ? Give an application for priority queue. [3]
(b) Write a 'C' program to reverse the given string using stack. Find out time complexity of program. [9]