Total No. of Questions—**8**] [Total No. of Printed Pages—**8**

Seat No.	

[4957]-1088

S.E. (I.T.) (II Sem.) EXAMINATION, 2016

DATA STRUCTURES AND FILES

(2012 PATTERN)

Time : Two Hours **Maximum Marks : 50**

N.B. **:—** (*i*) Answer *four* questions.

(*ii*) Neat diagrams must be drawn wherever necessary.

(*iii*) Figures to the right indicate full marks.

(*iv*) Assume suitable data, if necessary.

1. (*a*) Clearly indicate the contents of stack for evaluating the following postfix expression : [6]

Assume A = 10, B = 2, C = 13

(*i*) A B + C – B A – C + –

(*ii*) A B C + *C B A – + *

(*b*) Consider the following circular queue of character and size 5. [6]

0	1	2	3	4
		A	C	

Front point to index 1 and Rear point to index 3

For addition operation, first increment the corresponding index by 1 and then add the new element at the index location

P.T.O.

For deletion operation, first increment the corresponding index by 1 and then delete the element at the index location

Show the queue contents as per the following operations at every step :

(*i*) F is added to the queue

(*ii*) Two letters are deleted

(*iii*) K, L, M are added to the queue

(*iv*) Two letters are deleted

(*v*) R is added to the queue

(*vi*) Two letters are deleted.

Or

2. (*a*) Clearly indicate the contents of stack during conversion of given infix expression to prefix : [6]

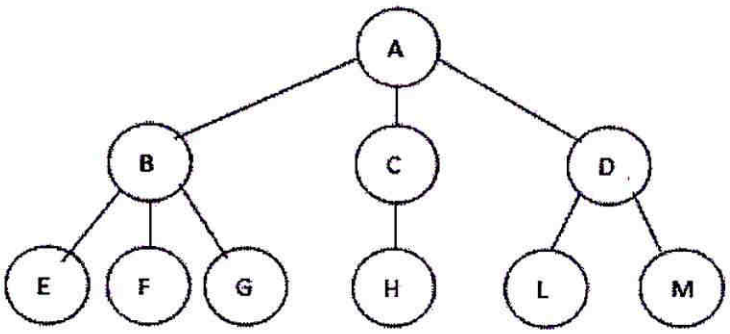
Infix expression : A + (B * ((D – E % F) / H))

(*b*) The dq is an input restricted doubly ended queue, implemented as a linear queue. The delete functions return the element that is deleted and also stores 0 (zero) at the location of deleted element. Clearly indicate the contents of queue after each add and delete operation given below : [6]

```
struct dqueue
{
    int front;
    int arr[10];
    int rear;
};
```

```
struct dqueue dq = {–1, {0}, –1};
int i = 0;
addqatend (&dq, 11);
addqatend (&dq, 12);
addqatend (&dq, 13);
i = delqatbeg (&dq);
i = delqatbeg (&dq);
i = delqatend (&dq);
addqatend (&dq, 22);
addqatend (&dq, 23);
i = delqatbeg (&dq);
i = delqatbeg (&dq);
addqatend (&dq, 24);
```

3. (*a*) Convert the following generalized tree into a binary tree : [4]

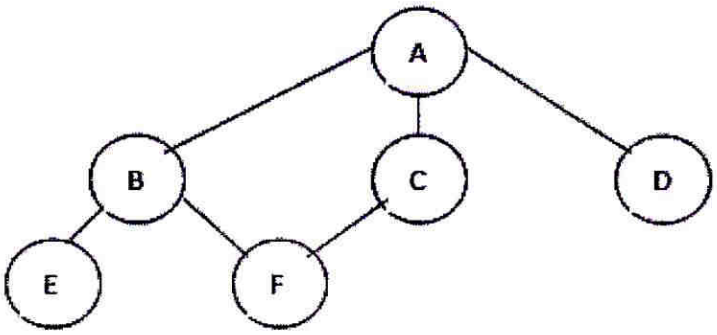


(*b*) Construct binary tree using tree traversals given below : [4]

Preorder traversal : P A Q B R S D E F

Inorder traversal : A P B Q D S E R F

(*c*) For the graph given below, find BFS and DFS. [4]



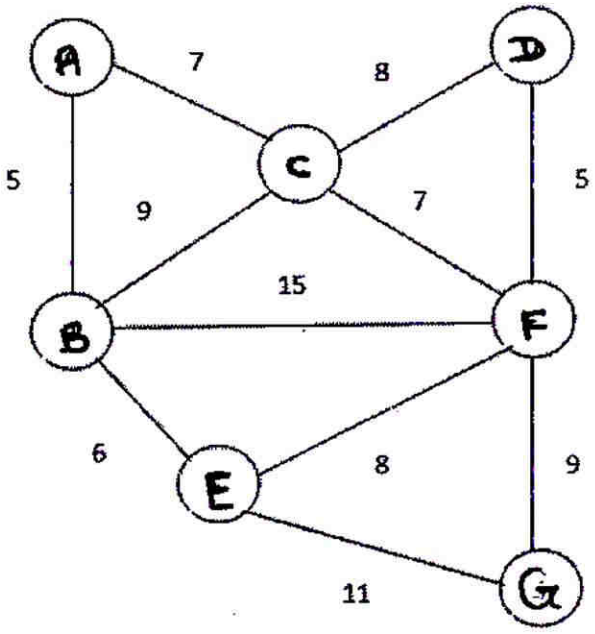
Or

4. (*a*) Draw the threaded binary tree equivalent for the tree represented by the following array assuming root node of tree is stored at index 0 in the array. [4]

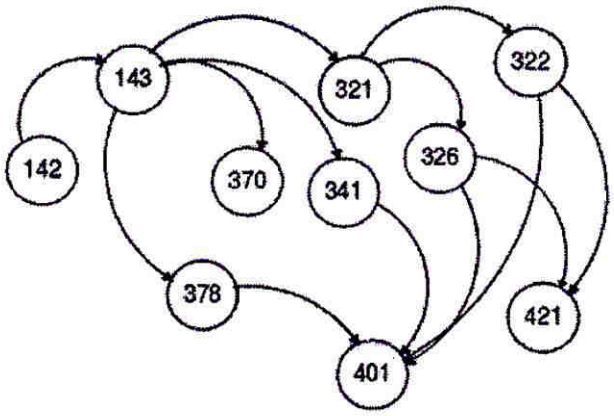
Index	**Data**
0	10
1	—
2	20
3	—
4	—
5	30
6	—
7	—
8	—
9	—
10	—
11	—
12	40
13	—
14	—
15	—
16	—

17	—
18	—
19	—
20	—
21	—
22	—
23	—
24	—
25	50
26	—
27	—
28	—
29	—
30	—
31	—

(*b*) For the graph given below, show stepwise representation of MST using Kruskal's algorithm. [4]



(*c*) Consider the following graph, each node contains subject course number : [4]



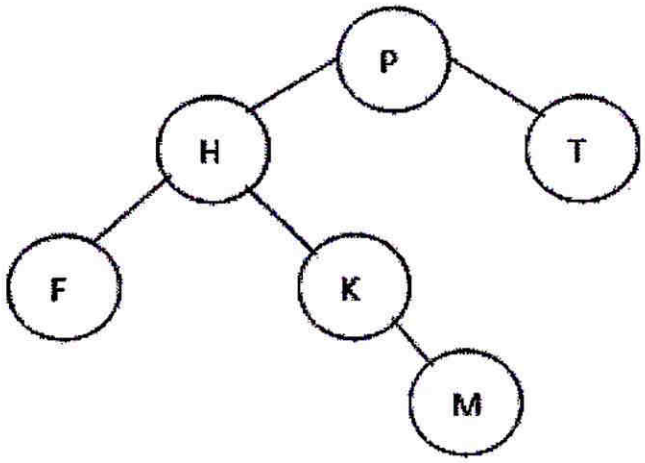
To complete a course, students need to take all the course subjects. Students need to take into consideration which courses are prerequisites for other courses when making a schedule for the upcoming semester so that 370 can not be taken before 143, but the former can be taken along with 341 and 370. Help the students by giving the order of subjects they need to take to complete the course. [Hint : Topological sort]

5. (*a*) Show stepwise contruction of maxheap for the data :

40, 50, 10, 60, 20, 30, 70 [5]

(*b*) What is symbol table ? What are the operations on symbol table. Give symbol table ADT. [5]

(*c*) Convert the following binary tree into AVL tree. [4]



Or

6. (*a*) Assume a hash table of size 10 and hash function :

H(X) = X mod 10

Perform linear probing with and without replacement for the given set of values : [6]

71, 63, 59, 20, 75, 105, 216, 89, 8, 29

(*b*) Construct optimal bianry search tree, for the following data : [8]

N = 4, Key set = {do, if, read, while}

{pl, p2, p3, p4} = (3, 3, 1, 1)

{q0, q1, q2, q4} = {2, 3, 1, 1, 1}

The p's and q's are multiplied by 16 for convenience.

7. (*a*) Explain how records are logically deleted from a file. [6]

(*b*) Explain 'C++' file open function with syntax, example. Explain the difference between 'C++' ios :: app and ios :: ate flags associated with file open function. [6]

Or

8. (*a*) Explain the following 'C' file functions with syntax and example : [6]

(*i*) Create file

(*ii*) Read (data and record from a file)

(*iii*) Delete file.

(*b*) Some N Employee records are stored in a sequentially organized file (emprec.dat). Write C++ code to find the value of N, without reading the records from the file one after another. [6]

```
struct Employee
{
        unsigned int empid;
        char fname[10];
        char lname[10];
        char gender;
        float salary;
};
```