

PR-601]

PR-601 Seat No. 1756

[Contd...

## B. C. A. (Sem. IV) Examination

April / May - 2016

BCA - 401 : Data Structure

<ol> <li>(a) Answer the following.</li> <li>(b) Define: Algorithm.</li> <li>(c) What is Primitive data structure?</li> <li>(d) Define: Algorithm.</li> <li>(e) Answer the following. (Any Two)</li> <li>(f) Explain Time and Space efficiency of an algorithm with example.</li> <li>(g) Explain Linear and Non-Linear data structure in detail.</li> <li>(g) Explain any two string manipulation functions with syntax and example.</li> <li>(e) Answer the following.</li> <li>(f) Compare stack and Queue.</li> <li>(g) Write short note on priority queue.</li> </ol>	Tim	e:3	Hou	ers] [Total Mark	s: <b>70</b>
<ul> <li>(2) What is Primitive data structure?</li> <li>(3) Define: Algorithm.</li> <li>(b) Answer the following. (Any Two)</li> <li>(1) Explain Time and Space efficiency of an algorithm with example.</li> <li>(2) Explain Linear and Non-Linear data structure in detail.</li> <li>(3) Explain any two string manipulation functions with syntax and example.</li> <li>(a) Answer the following.</li> <li>(b) Compare stack and Queue.</li> </ul>	1	(a)	Ans	wer the following.	6
<ul> <li>(3) Define: Algorithm.</li> <li>(b) Answer the following. (Any Two)</li> <li>(1) Explain Time and Space efficiency of an algorithm with example.</li> <li>(2) Explain Linear and Non-Linear data structure in detail.</li> <li>(3) Explain any two string manipulation functions with syntax and example.</li> <li>2 (a) Answer the following.</li> <li>(1) Compare stack and Queue.</li> </ul>			(1)	Define: Data Structure.	÷
<ul> <li>(b) Answer the following. (Any Two)</li> <li>(1) Explain Time and Space efficiency of an algorithm with example.</li> <li>(2) Explain Linear and Non-Linear data structure in detail.</li> <li>(3) Explain any two string manipulation functions with syntax and example.</li> <li>2 (a) Answer the following.</li> <li>(1) Compare stack and Queue.</li> </ul>			(2)	What is Primitive data structure?	
<ol> <li>Explain Time and Space efficiency of an algorithm with example.</li> <li>Explain Linear and Non-Linear data structure in detail.</li> <li>Explain any two string manipulation functions with syntax and example.</li> <li>Answer the following.</li> <li>Compare stack and Queue.</li> </ol>			(3)	Define: Algorithm.	
algorithm with example.  (2) Explain Linear and Non-Linear data structure in detail.  (3) Explain any two string manipulation functions with syntax and example.  2 (a) Answer the following.  (1) Compare stack and Queue.		(b)	Ans	wer the following. (Any Two)	12
structure in detail.  (3) Explain any two string manipulation functions with syntax and example.  2 (a) Answer the following.  (1) Compare stack and Queue.		-	(1)		n
functions with syntax and example.  2 (a) Answer the following.  (1) Compare stack and Queue.			(2)		a
(1) Compare stack and Queue.		ar	(3)		n
(1) Compare stack and Queue.					
	2	(a)	Answer the following.		5
(2) Write short note on priority queue.			(1)	Compare stack and Queue.	
			(2)	Write short note on priority queue.	

1

(b)	Answer the following. (Any two)					
	(1)	Write an algorithm to insert and remove an element from circular queue.				
	(2)	Write an algorithm to remove an element from Doubly linked list.				
	(3)	Explain the concept of multi-linked structure.				
(a)	Define the following terms.					
	(1)	Weighted graph.				
	(2)	Edge.				
	(3)	Loop.				
	(4)	Forest.				
	(5) Null graph.					
	(6)	Isolated node.				
(b)	b) Answer the following. (Any two)		12			
	(1)	Differentiate BFS and OFS. Also write an algorithm of BFS.				
	(2)	2) Write an algorithm to delete an element from binary search tree.				

(3)

3

element into a balanced tree?

Explain AUL tree. How do you insert an

4	(a)	Explain	the	following.
	` '	-		_

5

- (1) Hashing.
- (2) Linear search Techinque.
- (b) Answer the following. (Any two)

12

- (1) What is Collision? Explain any one method to resolve collision.
- (2) What is sorting? Write an algorithm to sort data using Radix sort.
- (3) Exlain Quick sort with suitable example.