

DATA STRUCTURES AND ALGORITHMS
Unit-III-Assignment

Reg. No. : RA2311026010677 Name : Namita Nahata

Year : 2nd

Branch: CSE AI/ML Section: AA-2

I. Choose the best answer:

1. Which of the following is not true for stacks?
 - (a) It is a linear data structure.
 - (b) It allows insertion/deletion of elements only at one end
 - (c) It is widely used by systems processes, such as compilation and program control
 - (d) It is based on First-In-First-Out principle
2. Which of the following is not an example of a stack?
 - (a) Collection of tiles one over another
 - (b) A set of bangles worn by a lady on her arm
 - (c) A line up of people waiting for the bus at the bus stop
 - (d) A pileup of boxes in a warehouse one over another
3. Tower of Hanoi can be regarded as a problem of which of the following data structures?
 - (a) Stack
 - (b) Queue
 - (c) Graph
 - (d) Tree
4. Recursive function calls are executed using which of the following data structures?
 - (a) Stack
 - (b) Queue
 - (c) Graph
 - (d) Tree
5. If 2, 1, 5, 8 are the stack contents with element 2 being at the top of the stack, then what will be the stack contents after following operations:

Push (11)
Pop ()
Pop ()
Pop ()
Push(7)

 - (a) 11, 2, 1
 - (b) 8, 11, 7
 - (c) 7, 5, 8
 - (d) 5, 8, 7

6. Which of the following is best suitable for storing a simple collection of employee records?

- (a)Stack
- (b)Queue
- (c)Array
- (d)None of the above

7. If 'top' points at the top of the stack and 'stack []' is the array containing stack elements, then which of the following statements correctly reflect the Push Operation for inserting 'item' into the stack?

- (a) $\text{top} = \text{top} + 1;$ $\text{stack}[\text{top}] = \text{item};$
- (b) $\text{stack}[\text{top}] = \text{item};$ $\text{top} = \text{top} + 1;$
- (c) $\text{stack}[\text{top}+1] = \text{item};$
- (d)Both (a) and (c) are correct

8. If 'top' points at the top of the stack and 'stack []' is the array containing stack elements, then which of the following statements correctly reflect the pop operation?

- (a) $\text{top} = \text{top} - 1;$ $\text{item} = \text{stack}[\text{top}];$
- (b) $\text{item} = \text{stack}[\text{top}];$ $\text{top} = \text{top} - 1;$
- (c) $\text{item} = \text{stack}[-\text{top}];$
- (d)Both (b) and (c) are correct

9. If a pop operation is performed on an empty stack, then which of the following situations will occur?

- (a)Overflow
- (b)Underflow
- (c)Array out of bound
- (d)None of the above

10. Which of the following is not a stack application?

- (a)Recursion control
- (b)Expression evaluation
- (c)Message queuing
- (d>All of the above are stack applications

11. Which of the following statements is not true for queues?

- (a)It is a linear data structure.
- (b)It allows insertion/deletion of elements only at one end.
- (c)It has two ends front and rear.
- (d)It is based on First-In-First-Out principle.

12. Which of the following statements is not an example of a queue?

- (a)Collection of tiles one over another.
- (b)A queue of print jobs.
- (c)A line up of people waiting for the bus at the bus stop.
- (d)All of the above are queue examples.

13. CPU sched
(a)Stack
(b)Queue
(c)Graph
 (d)Tree

14. W

13. CPU scheduler can be implemented by which of the following datastructures?

- (a)Stack
- (b)Queue
- (c)Graph
- (d)Tree

14. Which of the following is a type of a queue?

- (a)Circular queue
- (b)Priority queue
- (c)Double-ended queue
- (d)All of the above

15. If 1, 2, 3, 4 are the queue contents with element 1 at the front and 4 at the rear, then what will be the queue contents after following operations:

- Insert (5)
 - Delete ()
 - Delete ()
 - Delete ()
 - Insert (6)
 - Insert (-1)
 - Delete ()
- (a)5, 6, -1
 (b)4, 5, 6, -1
 (c)1, 2, 6
 (d)1, 2, 6, -1

16. Which of the following is best suitable for implementing a print scheduler?

- (a)Stack
- (b)Queue
- (c)Array
- (d)None of the above

17. If 'front' points at the front end of the queue, 'rear' points at the rear end of the queue and 'queue []' is the array containing queue elements, then which of the following statements correctly reflects the insert operation for inserting 'item' into the queue?

- (a)rear = rear + 1; queue [rear] = item;
- (b)front = front + 1; queue [front] = item;
- (c)queue [rear++] = item;
- (d)Both (a) and (c) are correct

18. If 'front' points at the front end of the queue, 'rear' points at the rear end of the queue and 'queue []' is the array containing queue elements, then which of the following statements correctly reflects the delete operation for deleting an element from the queue?

- (a)item = queue [rear]; rear = rear + 1;
- (b)item = queue [front]; front = front + 1;
- (c)item = queue [++front];
- (d)Both (b) and (c) are correct

19. If a delete operation is performed on an empty queue, then which of the following situations will occur?
- Overflow
 - Underflow
 - Array out of bound
 - None of the above

20. Which of the following is not a queue application?
- Recursion control
 - CPU scheduling
 - Message queuing
 - All of the above are queue applications

ANSWERS

1.	d	2.	c	3.	b	4.	a	5.	c	6.	d	7.	a	8.	b	9.	b	10.	c
11.	b	12.	a	13.	b	14.	d	15.	a	16.	b	17.	d	18.	b	19.	b	20.	a

II. Place me in the basket:

Stack	Queue	Double Ended Queue	Notations
push	Enqueue	DequeueFront	Infix
pop	Dequeue	EnqueueFront	Prefix
peek		DequeueRear	Postfix
		EnqueueRear	

Following words are to be placed in the relevant basket:

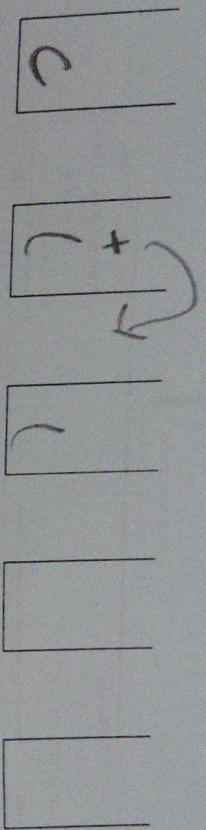
Push	Enqueue	Infix	EnqueueFront
Pop	Dequeue	DequeueRear	Prefix
Peek	DequeueFront	EnqueueRear	Postfix

III. Match me:

Column A	Column B
Infix	+/ABC 2
Prefix	AB/C+ 3
Postfix	A/B+C 1

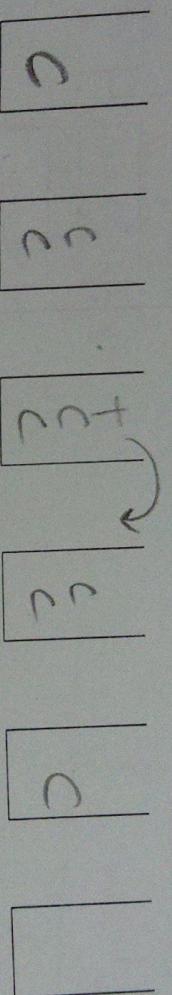
IV. Balance me:

(a+b)



Stack is \rightarrow a b +

((a+b))

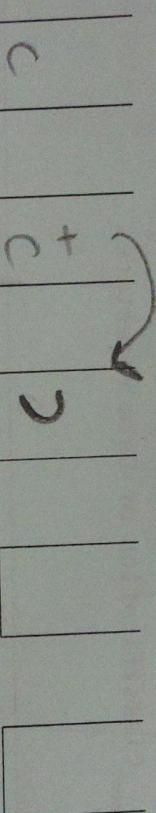


Stack is \rightarrow

ab +

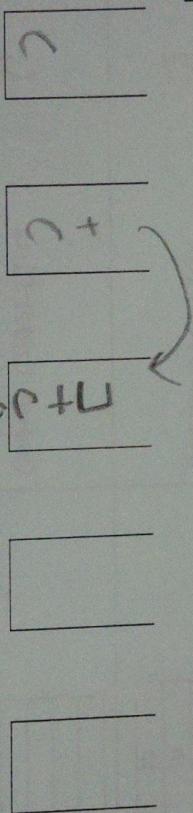
Expression is unbalanced.

(a+b))



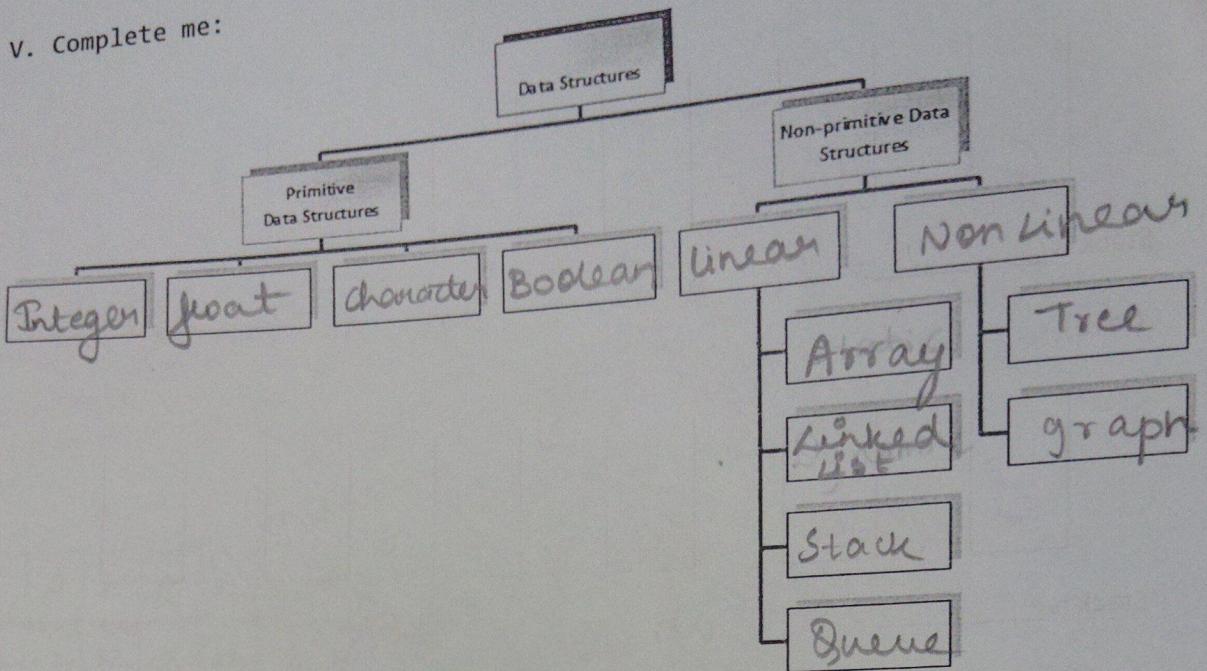
Stack is unbalanced ab +

(a+b]



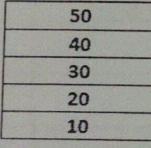
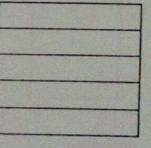
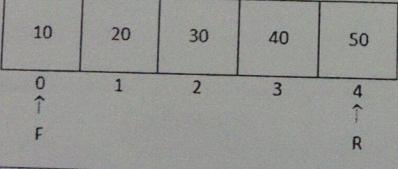
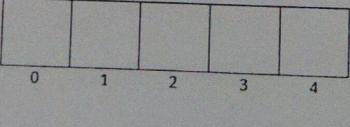
Not corresponding to the expression ab \rightarrow ab

V. Complete me:



Classification of Data Structures

VI. Match me:

Column A	Column B
Top → 	Queue Overflow 3
 Top →	Queue Underflow 4
	Stack Overflow 1
 None ↑ F R	Stack Underflow 2

VII. Complete me:

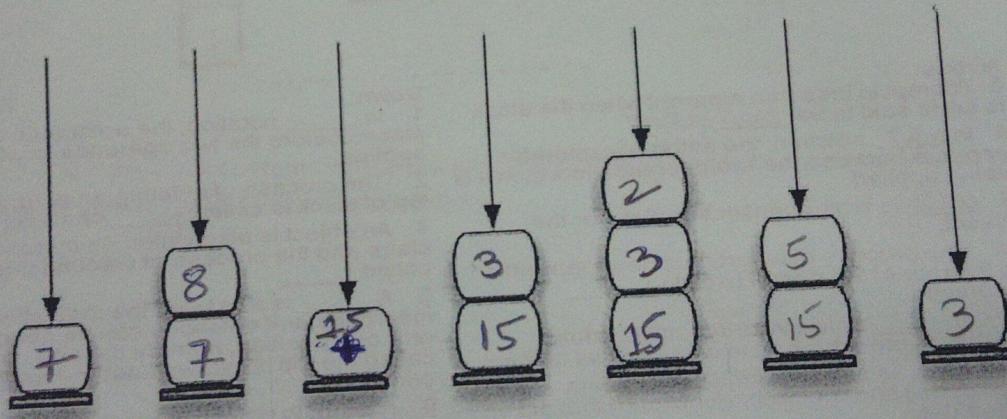
Push(50)	Enqueue(50)	Enqueue(50)																									
<p>Top →</p> <table border="1"> <tr><td>50</td></tr> <tr><td>40</td></tr> <tr><td>30</td></tr> <tr><td>20</td></tr> <tr><td>10</td></tr> </table>	50	40	30	20	10	<table border="1"> <tr><td>10</td><td>20</td><td>30</td><td>50</td><td></td></tr> <tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr> <tr><td>↑</td><td></td><td>↑</td><td></td><td></td></tr> <tr><td>F</td><td></td><td>R</td><td></td><td></td></tr> </table>	10	20	30	50		0	1	2	3	4	↑		↑			F		R			<p>queue [0] queue [1] queue [2] queue [3] queue [4] queue [5] queue [6] queue [7]</p> <p>Rear queue [0] queue [1] queue [2] 50 30 23 queue [3] queue [4] queue [5] queue [6] queue [7] queue [6] queue [5] queue [4] queue [3] queue [2] queue [1] queue [0] 23 9 12 1 2 50 30 Front</p>
50																											
40																											
30																											
20																											
10																											
10	20	30	50																								
0	1	2	3	4																							
↑		↑																									
F		R																									

VIII. Complete me:

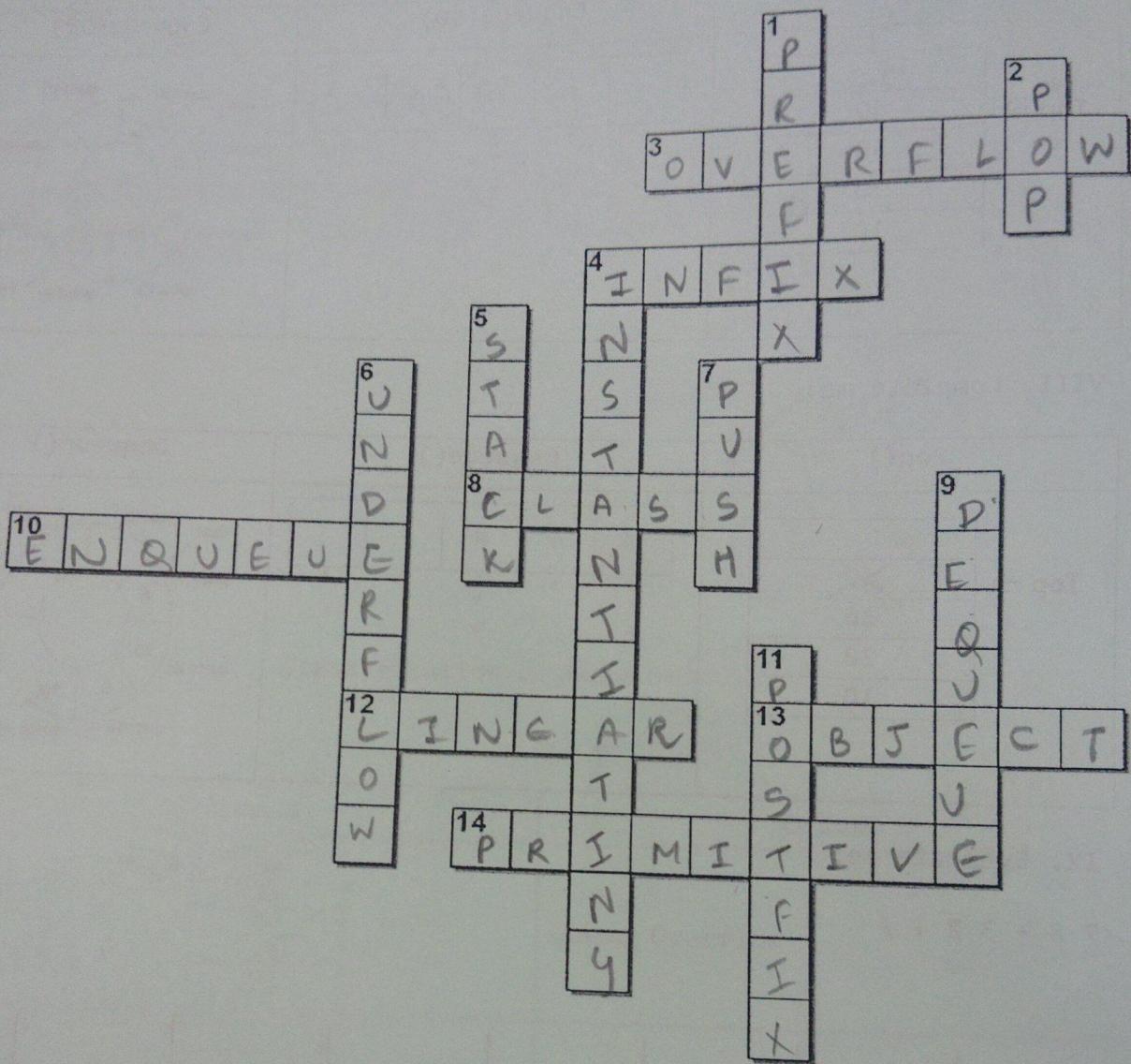
Pop()	Dequeue()	Dequeue()																								
<p>Top →</p> <table border="1"> <tr><td>40</td></tr> <tr><td>30</td></tr> <tr><td>20</td></tr> <tr><td>10</td></tr> </table>	40	30	20	10	<table border="1"> <tr><td>X</td><td>20</td><td>30</td><td></td><td></td></tr> <tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr> <tr><td>↑</td><td></td><td>↑</td><td></td><td></td></tr> <tr><td>F</td><td></td><td>R</td><td></td><td></td></tr> </table>	X	20	30			0	1	2	3	4	↑		↑			F		R			<p>queue [0] queue [1] queue [2] queue [3] queue [4] queue [5] queue [6] queue [7]</p> <p>Rear queue [0] queue [1] queue [2] X 30 23 queue [3] queue [4] queue [5] queue [6] queue [7] queue [6] queue [5] queue [4] queue [3] queue [2] queue [1] queue [0] 23 9 12 1 2 X 30 Front</p>
40																										
30																										
20																										
10																										
X	20	30																								
0	1	2	3	4																						
↑		↑																								
F		R																								

IX. Evaluate me:

7 8 + 3 2 + /



X. Fill me:



Across:

- 3 Attempt to insert an element when the stack is full is said to be overflow.
- 4 In infix notation, the arithmetic operator appears between the two operands to which it is being applied.
- 8 class is a blueprint or template for the object.
- 10 The process of inserting a new element on to the rear of the queue is called _____.
- 12 In _____ data structures, all the data elements are arranged in a sequential fashion.
- 13 object is simply a collection of data (variables) and methods (functions) that act on those data.
- 14 _____ data structures include all the fundamental data structures that can be directly manipulated by machine level instructions.

Down:

- 1 In _____ notation, the arithmetic operator is placed before the two operands to which it applies.
- 2 The process of deleting an element from the top of stack is called pop operation.
- 4 An object is also called an instance of a class and the process of creating this object is called _____.
- 5 A _____ is a list with the restriction that insertions and deletions can be performed in only one position, namely, the end of the list, called the top. It follows Last-In-First-Out (LIFO) principle.
- 6 Attempt to delete an element when the stack is empty is said to be _____.
- 7 The process of inserting a new element to the top of the stack is called push operation.
- 9 The process of deleting an element from the front of queue is called _____ operation.
- 11 In _____ notation, the arithmetic operator appears directly after the two operands to which it applies.