

Data Structures and Algorithms

Assignment 03

Marks 35

Instructions

Work on this assignment individually.

Absolutely NO collaboration is allowed. Any traces of plagiarism would result in a ZERO marks in this assignment and possible disciplinary action.

Solve all the questions carefully, you can solve them on a simple page instead of taking printouts of this document.

Only **handwritten solution** will be accepted.

Attach your document (image, PDF or in any other digital format) with an Email and send it to the following respective recipient till **Monday, April 12, 2021**.

DO NOT compress/zip your solution.

The email must be sent from your **official PUCIT email id**, otherwise it will not be accepted and will be marked **ZERO**.

The subject of the email should be the exact name of the **assignment** i.e. **Assignment 03**.

Degree	Recipient Email	Subject of Email
BSIT Morning	dsaubt03@gmail.com	Assignment 03
BSIT Afternoon	dsaubt04@gmail.com	

5 MARKS will be DEDUCTED if submission instructions are not followed.

No submissions will be considered after due date.

Name: _____

Roll #: _____

Question # 01

[05]

Trace following code by showing contents after each call in sequence of the initially **empty circular queue q** having **max size of 3**:

Call No.	Call	Array based circular queue's content after the execution of call		
1	q.enqueue('A');			
2	q.enqueue('B');			
3	q.enqueue('C');			
4	q.dequeue();			
5	q.dequeue();			
6	q.enqueue('D');			
7	q.enqueue('E');			
8	q.dequeue();			
9	q.enqueue('F');			
10	q.dequeue();			

Question # 02

[05]

Convert the following **infix** expression into its **prefix** form without using the **stack**.

$$((A + ((B - C) + D)) * (E - (A/C)))$$

Question # 03

[05]

Convert the following **infix** expression into its **postfix** form without using the **stack**.

$$((A + ((B - C) + D)) * (E - (A/C)))$$

Question # 04

[05]

Use a **stack** to **transform** the following **infix** expression into its **prefix** form. Show clearly the **output stream**, **stack**, and the most recently **read character**. Show the **stack** snapshot whenever it changes.

$$A * (B - (C + D) / E)$$

Question # 05

[05]

Use a **stack** to **transform** the following **infix** expression into its **postfix** form. Show clearly the **output stream**, **stack**, and the most recently **read character**. Show the **stack** snapshot whenever it changes.

$$A * (B - (C + D) / E)$$

Question # 06

[05]

Use a **stack** to **evaluate** the following **prefix** expression. Show clearly the **output stream**, **stack**, and the most recently **read character**. Show the **stack** snapshot whenever it changes.

$$- + * - 5 7 / 4 2 3 9$$

Question # 07

[05]

Use a **stack** to **evaluate** the following **postfix** expression. Show clearly the **output stream**, **stack**, and the most recently **read character**. Show the **stack** snapshot whenever it changes.

$$7 5 2 + * 4 1 5 - / -$$

NOTE: - No submission will be accepted after the DUE DATE.

B E S T O F L U C K