



CET 214 - Data Structures & Algorithms

Experiment # 4

Experiment Title

Queue Operations

Assessment of CLO(s): IV

Performed on 11-10-2024

Student Name			
Roll No.		Group	
Semester		Session	

Total (Max)	Criteria 1 (2.5)	Criteria 2 (2.5)	Criteria 3 (2.5)	Criteria 4 (2.5)	Total (10)
Marks Obtained					
Remarks (if any)					

Experiment evaluated by

Instructor's Name	Engr. Muhammad Asad Husain		
Date		Signature	

Department of Engineering Technology
(UIT University)

Course Code: CET214 Course Title: Data Structures & Algorithms Credits: 2+1 Session: Fall 2024

Rubric for assessment criteria to perform experiment number 4.

Level Criteria	UNSATISFACTORY 1	COMPETENT 2	PROFICIENT 3	DISTINGUISHED 4
Capability of writing algorithm/ Procedure	None of the steps are implemented of an algorithm.	Few steps are implemented correctly of an algorithm.	Most of the steps are implemented correctly of an algorithm.	All the steps are implemented correctly of an algorithm.
Capability of writing Program	Programs not completed.	Completeness of code, consistent variable naming and unformatted.	Completeness of code, inconsistent variable naming and well formatted.	Completeness of code, consistent variable naming and well formatted.
Completion of target in Lab	25% target has been completed	50% target has been completed	75% target has been completed	100% target has been completed
Output	None of the outputs are correct.	Few outputs have been found correctly.	Some of the outputs are correct and well formatted.	Most of the outputs are correct and well formatted.

Practical Objective(s):

1. To implement Queues on Linear Arrays.
2. Insertion and Deletion in queues.

Theory**Queue**

A queue is a linear list of elements in which insertions can take place only at one end, called the rear, and deletions can take place only at one end, called the front. Queues are also called First In First Out (FIFO) lists, since the first element in a queue is the first element out of the queue. In other words, the order in which elements enter a queue is the order in which they leave. This contrasts with stacks, which are Last In First Out (LIFO) lists.

Queues are found in everyday life. The automobiles waiting to pass through an intersection form a queue, in which the first car in line is the first car through; the people waiting in line at a bank form a queue.

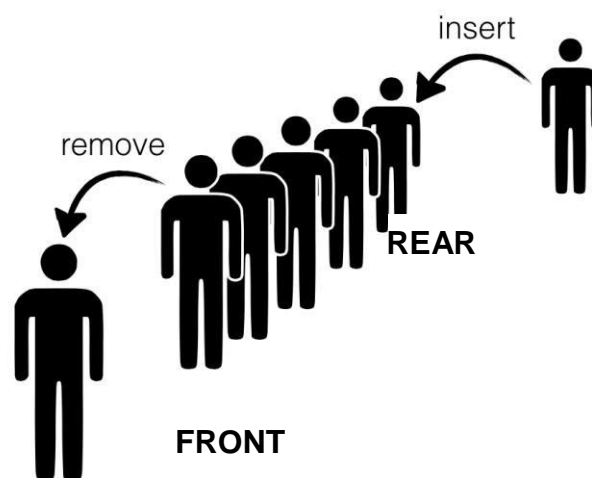


Figure 1. Real world example of a Queue

We will now discuss two basic operations associated with queues:

1. INSERT

In a queue, insertion can take place only at the REAR end of the queue. Whenever an element is added to the queue, the value of REAR is increased by 1.

2. DELETE

Deletion can take place only from the FRONT of a queue. Whenever an element is deleted from a queue, the value of the FRONT is increased by 1.

Example:

AAA	BBB	CCC				...	
0	1	2	3	4	5		N

Front: 0

Rear: 2

AAA	BBB	CCC	DDD			...	
0	1	2	3	4	5		N

Front: 0

Rear: 3

	BBB	CCC	DDD			...	
0	1	2	3	4	5		N

Front: 1

Rear: 3

Figure 2. Queue Insert/Delete Operation**Procedure 1: QINSERT (QUEUE, N, FRONT, REAR, ITEM)**

This procedure inserts an element ITEM into a QUEUE.

Step # 1: [Queue already filled?]

If FRONT=0 and REAR=N or If FRONT=REAR+1, then:

Write: OVERFLOW and Return.

Step # 2: [Find new value of REAR]

If FRONT: =NULL, then [Queue initially empty]

Set FRONT: =0 and REAR: =0

Else:

Set REAR = REAR+1

[End of IF Structure]

Step # 3: Set QUEUE [REAR]: = ITEM. [This inserts new element]**Step # 4: Return.**

Procedure 2: QDELETE (QUEUE, N, FRONT, REAR, ITEM)

This procedure deletes an element from a QUEUE and assigns 0 in its place.

Step # 1: [Queue already empty?]

If FRONT: =NULL, then:

Write: UNDERFLOW and Return.

Step # 2: Set QUEUE [FRONT]: = 0. [Assigns 0 at deleted element]**Step # 3: [Find new value of FRONT]**

If FRONT: =REAR, then [Queue has only one element to start]

Set FRONT: =NULL and REAR: =NULL

Else:

Set FRONT = FRONT+1

[End of If Structure]

Step # 4: Return.**Do It Yourself:**

1. Create two functions qinsert() and qdelete() by implementing procedure 1 and procedure 2, respectively .
2. Write a program which inserts five students' names in a queue and then deletes 2 names from that queue using qinsert() and qdelete().