

Mukesh Patel School of Technology Management & Engineering
Computer Engineering Department
Program: B.Tech Integrated. Sem V

Course: Basic Data Structures

Experiment No.05

PART A

(PART A : TO BE REFERRED BY STUDENTS)

A.1 Aim:

To study and implement concept of queue data structure.

TASK 1:

Write a Program to implement ADT of Linear Queue

A.2 Prerequisite:

1. Knowledge of different operations performed on Queue data structure
2. Fundamental concepts of C\C++.

A.3 Outcome:

After successful completion of this experiment students will be able to

1. Identify the steps of queue data structure selection.
2. Implement Queue data structure to solve the given problem
3. Enlist the applications of stack data structure.
4. Differentiate between different types of Queue.

A.4 Theory:

A.4.1. Introduction of Queue

A.5 Procedure/Algorithm:

A.5.1:

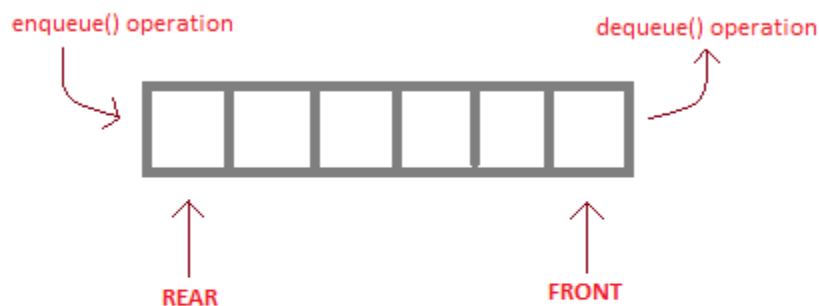
Queue is a linear data structure in which items are inserted at one end called 'Rear' and deleted from the other end called 'Front'. Queues are based on the First-In-First-Out (FIFO) principle that means the data item that is inserted first in the queue is also the first one to be removed from the queue.

A queue or FIFO (first in, first out) is an abstract data type that serves as a collection of elements, with two principal operations: enqueue, the process of adding an element to the collection. (The element is added from the rear side) and dequeue, the process of removing the first element that was added. (The element is removed from the front side). It can be implemented by using both array and linked list.

Representation of Queue:

1. Static representation using arrays.
2. Dynamic representation using linked list.

Array representation of a queue needs two indices: FRONT and REAR.



enqueue() is the operation for adding an element into Queue.

dequeue() is the operation for removing an element from Queue .

QUEUE DATA STRUCTURE

Array declaration in C++ as follows: `int queue[maxsize]`

Conditions to be assumed:

1. $\text{FRONT} = \text{REAR}$ or $\text{FRONT} = \text{REAR} = -1$, if the queue empty
2. Whenever an element is deleted from the queue, $\text{FRONT} = \text{FRONT} + 1$
3. Whenever an element is inserted into the queue, $\text{REAR} = \text{REAR} + 1$

Operations on Queue:

Operation	Description	Restriction
Create Queue	It creates a new empty queue. This operation must be carried out in order to make the queue logically accessible	-
Qinsert	It inserts a new data item at the rear of the queue	Queue must not be full
Qdelete	It deletes and then return the data item at the front of the queue.	Queue must not be empty
Queue full	It returns true if the queue is full. Otherwise it returns false.	-
Queue Empty	It returns true if the queue is empty. Otherwise it returns false.	

PART B**(PART B : TO BE COMPLETED BY STUDENTS)**

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case there is no Black board access available)

Roll No.	Name:
Class :	Batch :
Date of Experiment:	Date of Submission
Grade :	Time of Submission:
Date of Grading:	

B.1 Software Code written by student: (Task 1)

(Paste your code completed during the 2 hours of practical in the lab here)

Task1:

Task2:

B.2 Input and Output: (Task 1)

(Paste your program input and output in following format, If there is error then paste the specific error in the output part. In case of error with due permission of the faculty extension can be given to submit the error free code with output in due course of time. Students will be graded accordingly.)

Task1:

B.3 Observations and learning [w.r.t. all tasks]:

(Students are expected to comment on the output obtained with clear observations and learning for each task/sub part assigned)

B.4 Conclusion:

(Students must write the conclusion as per the attainment of individual outcome listed above and learning/observation noted in section B.3)

B.5 Question of Curiosity

(To be answered by student based on the practical performed and learning/observations)

Discuss the limitation of linear queue and elaborate any three applications of queue.
