**East West University**

**Department of Computer Science & Engineering**

**A/2, Jahurul Islam Avenue, Jahurul Islam City, Aftabnagar, Dhaka-1212**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Lab Manual : 04(Queue)**

**Course Code : CSE207**

**Course Title : Data Structure**

**Instructor : Tanni Mittra, lecturer, Department of CSE**

**Objective:**

At first, create an ADT of the queue that will contain the below function:

* enqueue
* dequeue
* Size
* Isempty
* Destroy
* Top

After implementing the ADT solve the following problems by using the required functions from the ADT.

**Lab Task:**

**Exercise 1:**

**CopyQueue**

Write a program that copies the content one queue to another

|  |  |
| --- | --- |
| Input Data | Output Data |
| Q1: 1 2 3 4 5 | Q2: 1 2 3 4 5 |

**Exercise 2:**

**Categorize elements in Queue**

Write a program that creates a Queue and the categorize elements in the queue in such way that numbers less than 18 will be in group1, numbers between 18 to 35 will be ingroup2, numbers between 36 to 45 will be in group3, and number greater that 46 will be in group4.

|  |  |
| --- | --- |
| Input Data | Output Data |
| Q: 12 25 38 45 5 | Group1: 12 5  Group2:25 38  Group3:45  Group4: NULL |

**Exercise 3:**

**Delete all Negative Integer**

Write a program that will take a queue of integers and deletes all negative integers without changing the order of the remaining elements in queue.

|  |  |
| --- | --- |
| Input Data | Output Data |
| Q: 1 2 -3 4 - 5 | Q: 1 2 4 |

**Exercise 4:**

Write an algorithm called stackToQueue that creates a queue from a stack. After the queue has been created, the top of the stack should be the front of the queue and the base of the stack should be the rear of the queue. At the end of the algorithm, the stack should be empty.

**Exercise 5:**

Write an algorithm that reverses the contents of a queue.

**Exercise 6:**

Write an algorithm called queueToStack that creates a stack from a queue. At the end of the algorithm, the queue should be unchanged; the front of the queue should be the top of the stack, and the rear of the queue should be the base of the stack.