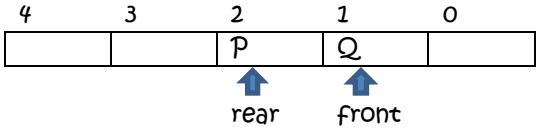
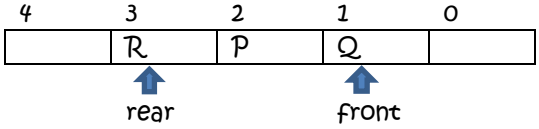
QUEUE DATA STRUCTURE (TUTORIAL 2)

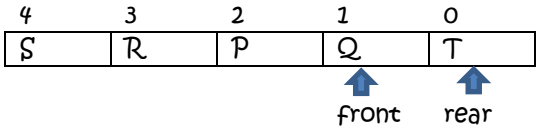
Question 1

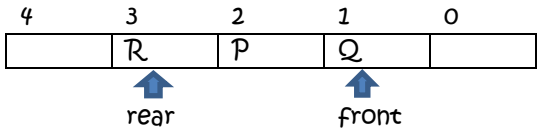
a) Consider the following Circular Queue and draw the queue frames after executing each statement given below.

i) insert('R');
ii) peekFront();
iii) insert('S');
iv) insert('T');
v) remove();

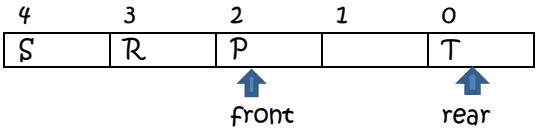




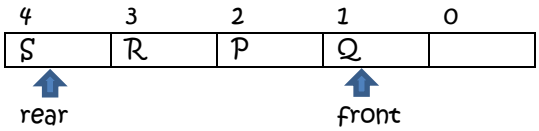




returns Q



returns Q



b) What will happen if the above queue is a linear queue?

Can't insert T

Question 2

i) Assume that a queue class has already been implemented to store double values. Write an application to insert 5 numbers from the keyboard to a queue object created from the class.
ii) Modify the application to retrieve values from the queue and print them.
iii) Comment on the order of insertion to the queue and the order of retrieval from the queue.

```
import java.util.Scanner;
class QueueM{
	public static void main(String[] args) {
		Scanner sc = new Scanner(System.in);
		QueueX q = new QueueX(5);

		System.out.println("Enter values: ");

		for(int i = 0; i < 5; i++) {
			double d = sc.nextDouble();
			q.insert(d);
		}

		while(!q.isEmpty()) {
			System.out.print(q.remove() + " ");
		}

	}
}
//input: 5 6 7 8 9
// output: 5.0 6.0 7.0 8.0 9.0
```

Question 3

Write a program to reverse the first k elements of a given queue. Assume the queue class is available with insert(), remove() and peek() methods.

Hint: Use a stack.

```
import java.util.Scanner;
public class QueueMain {

	public static void main(String[] args) {
		Scanner sc = new Scanner(System.in);
		QueueX qx = new QueueX(10);
		StackX sx = new StackX(5);

		System.out.println("Enter values: ");

		for(int i = 0; i < 5; i++) {
			int value = sc.nextInt();
			qx.insert(value);
		}

		System.out.println("How many elements do you wish to reverse? ");
		int k = sc.nextInt();

		for(int i = 0; i < k; i++) {
			sx.push(qx.remove());
		}

		while (!sx.isEmpty()) {
			qx.insert(sx.pop());
		}

		for (int i = 0; i < 5-k; i++) {
			qx.insert(qx.remove());
		}

		while(!qx.isEmpty()) {
		System.out.print(qx.remove() + " ");
		}


	}//main

}//class
```

Additional Exercises:

Question 1

i) You are required to build the following class in your program.

printerLine
- jobArr[]
- insert(int jobID) - remove() - isEmpty() - isFull()

A printer processes the jobs that are sent to be printed in the same order as it receives. You are required to create an application which will do the same function that a printer does. (printer queue)

Hint: In addition to the attributes and methods given above you can use the necessary attributes and methods that are related with the data structure you have selected

a) Implement the constructor printerLine(int size);
b) Implement insert(), remove(), isEmpty() and isFull() methods of the class.
c) Write a main Program to create an object with 5 elements of the printerLine class.
d) Allow the user to input 5 JobIDs from the keyboard.

Enter JobID: 101
Enter JobID: 22
Enter JobID: 180
Enter JobID: 111
Enter JobID: 50

e) You are required to send JobIDs to separate PCs: Jobs sent to PC1 are even numbers and jobs sent to PC2 are odd numbers.
(Ex: JobID 22 is send to PC1 and JobID 111 is send to PC2)
f) Write the code to remove the jobs and display the result as follows.
JobID 22 (PC1)
JobID 111 (PC2)

```
public class PrinterLine {

	private int maxSize;   //max number of locations
	private int[] jobArray;   //array definition
	private int front, rear, noOfItems;  //index definitions

	public PrinterLine(int size) {   //constructor
		maxSize = size;
		jobArray = new int[maxSize]; //array implementation
		front = noOfItems = 0;
		rear = -1;
	}//constructor

	public void insert(int j) {   //insert method
		if(rear == maxSize-1) {
			System.out.println("Queue overflow");
		}
		else {
			jobArray[++rear] = j;
			noOfItems++;
		}
	}//insert

	public int remove() {   //remove method
		if(noOfItems == 0) {
			System.out.println("Queue underflow");
			return 0;
		}
		else {
			noOfItems--;
			return jobArray[front++];
		}
	}//remove

	public boolean isEmpty() {
		return (noOfItems == 0);
	}
```

```
	public boolean isFull() {
		return (rear == maxSize-1);
	}

}//class


import java.util.Scanner;
public class QueueMain {

	public static void main(String[] args) {
		Scanner sc = new Scanner(System.in);
		PrinterLine pl = new PrinterLine(5);

		for(int i = 0; i < 5; i++) {
			System.out.print("Enter Job ID: ");
			pl.insert(sc.nextInt());
		}

		while(!pl.isEmpty()) {
			int temp = pl.remove();
			if(temp % 2 == 0) {
				System.out.println("Job ID " + temp + "\t(PC1)");
			}
			else
				System.out.println("Job ID " + temp + "\t(PC2)");
		}

	}//main
}//class
```