1. Implement an ArrayDequeue and all of its methods such as add(), addFirst(), addLast(), element(), poll(), push(), remove.

//Implement a PriorityQueue and use all the methods.

package SBA\_3;

import java.util.Iterator;

import java.util.PriorityQueue;

public class Q2 {

// Main Method

public static void main(String args[])

{

PriorityQueue<String> pq = new PriorityQueue<>();

pq.add("Ajay");//A-Z= 65-90

pq.add("Vijay");

pq.add("Raj");

pq.add("Gagan");

System.out.println("head:"+pq.element());

System.out.println("head:"+pq.peek());

System.out.println("iterating the queue elements:");

Iterator itr=pq.iterator();

while(itr.hasNext()){

System.out.println(itr.next());

}

pq.remove();//removes the head element

pq.poll(); //removes the head

System.out.println("after removing two elements:");

Iterator<String> itr2=pq.iterator();

while(itr2.hasNext()){

System.out.println(itr2.next());

}

}

}

2. Implement a PriorityQueue and use all the methods.

3. Implement a Stack and all of its methods peek(), push(), pop(), and to determine the size of the stack.

4. Write a program to implement insertion sort.

//Write a program to implement insertion sort.

**package** SBA\_3;

**public** **class** Q4 {

/\*Function to sort array using insertion sort\*/

**void** sort(**int** arr[])

{

**int** n = arr.length;

**for** (**int** i = 1; i < n; ++i) {

**int** key = arr[i];

**int** j = i - 1;

/\* Move elements of arr[0..i-1], that are

greater than key, to one position ahead

of their current position \*/

**while** (j >= 0 && arr[j] > key) {

arr[j + 1] = arr[j];

j = j - 1;

}

arr[j + 1] = key;

}

}

/\* A utility function to print array of size n\*/

**static** **void** printArray(**int** arr[])

{

**int** n = arr.length;

**for** (**int** i = 0; i < n; ++i)

System.***out***.print(arr[i] + " ");

System.***out***.println();

}

// Driver method

**public** **static** **void** main(String args[])

{

**int** arr[] = { 12, 11, 13, 5, 6 };

Q4 ob = **new** Q4();

ob.sort(arr);

*printArray*(arr);

}

}