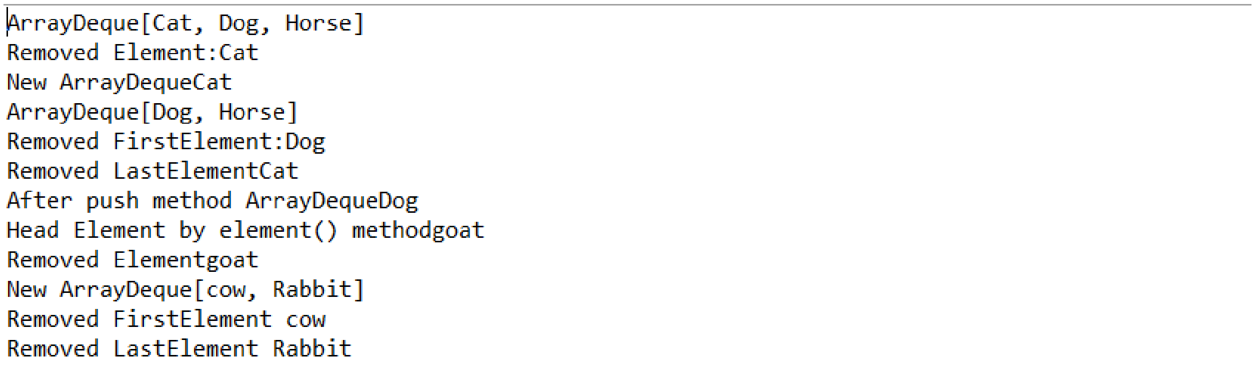
1.Implement an ArrayDequeue and all of its methods such as add(), addFirst(), addLast(), element(), poll(), push(), remove.

```
package sba3;
import java.util.ArrayDeque;
import java.util.Deque;
public class Ques1{
public static void main(String[] args) {
      ArrayDeque<String> animals=new ArrayDeque<>();
      //using add
      animals.add("Dog");
      //using addfirst
      animals.addFirst("Cat");
      //using addLast
    animals.addLast("Horse");
    System.out.println("ArrayDeque"+animals);
      //Using Poll()
    String element=animals.poll();
      System.out.println("Removed Element:"+element);
      System.out.println("New ArrayDeque"+element);
      System.out.println("ArrayDeque"+animals);
      //Using PollFirst()
    String firstelement=animals.pollFirst();
      System.out.println("Removed FirstElement:"+firstelement);
      //Using PollLast()
       String Lastelement=animals.pollLast();
      System.out.println("Removed LastElement"+element);
      //Using push()
    animals.push("Rabbit");
    animals.push("cow");
    animals.push("goat");
    System.out.println("After push method ArrayDeque"+firstelement);
    //Using elemnt()--returns element present in the head
    System.out.println("Head Element by element() method"+animals.element());
    //Using remove()
    String Element1=animals.remove();
    System.out.println("Removed Element"+Element1);
    System.out.println("New ArrayDeque"+animals);
    //Using removeFirst()
    String FirstElement1=animals.removeFirst();
    System.out.println("Removed FirstElement "+FirstElement1);
    String lastElement1=animals.removeLast();
    System.out.println("Removed LastElement "+lastElement1);

}

}
```

//Output



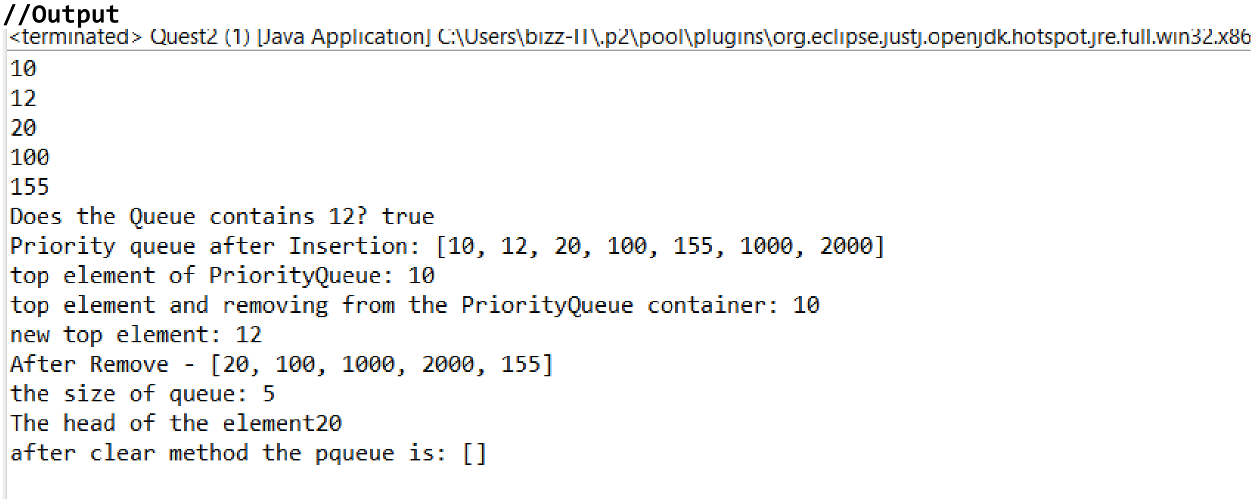
```
ArrayDeque[Cat, Dog, Horse]
Removed Element:Cat
New ArrayDequeCat
ArrayDeque[Dog, Horse]
Removed FirstElement:Dog
Removed LastElementCat
After push method ArrayDequeDog
Head Element by element() methodgoat
Removed Elementgoat
New ArrayDeque[cow, Rabbit]
Removed FirstElement cow
Removed LastElement Rabbit
```

2. Implement a PriorityQueue and use all the methods.

```java
package sba3;

import java.util.Iterator;
import java.util.PriorityQueue;



public class Quest2 {

public static void main(String[] args) {
              // Creating empty priority queue
              PriorityQueue<Integer> pQueue = new PriorityQueue<Integer>();
              // Adding items to the pQueue using add()
              pQueue.add(10);
              pQueue.add(12);
              pQueue.add(20);
              pQueue.add(100);
              pQueue.add(155);
              System.out.println("the priority queue: " + pQueue);
              // Creating an iterator
              Iterator <Integer>value =pQueue.iterator();
              // Displaying the values after iterating through the queue
              System.out.println("The iterator values are: ");
              while (value.hasNext()) {
              System.out.println(value.next());
              }
              // Check for "4" in the queue
              System.out.println("Does the Queue contains 12? "+pQueue.contains(12));
              // Inserting using offer()
              pQueue.offer(1000);
              pQueue.offer(2000);
              // Displaying th final Queue
              System.out.println("Priority queue after Insertion: " +pQueue );
              // Printing the top element of PriorityQueue
              System.out.println("top element of PriorityQueue: " + pQueue.peek());
              // Printing the top element and removing it
              // from the PriorityQueue container
              System.out.println("top element and removing from the PriorityQueue container: " + pQueue.poll());
```

```
        // Printing the top element again
        System.out.println("new top element: " + pQueue.peek());
        // using the method
        pQueue.remove(12);
        System.out.println("After Remove - " + pQueue);
        //to find size
        System.out.println("the size of queue: "+pQueue.size());
        //element()
        System.out.println("The head of the element"+pQueue.element());
        // Creating an iterator
        //clear()
        pQueue.clear();
        System.out.println("after clear method the pqueue is: "+pQueue);
        }}
```

//Output



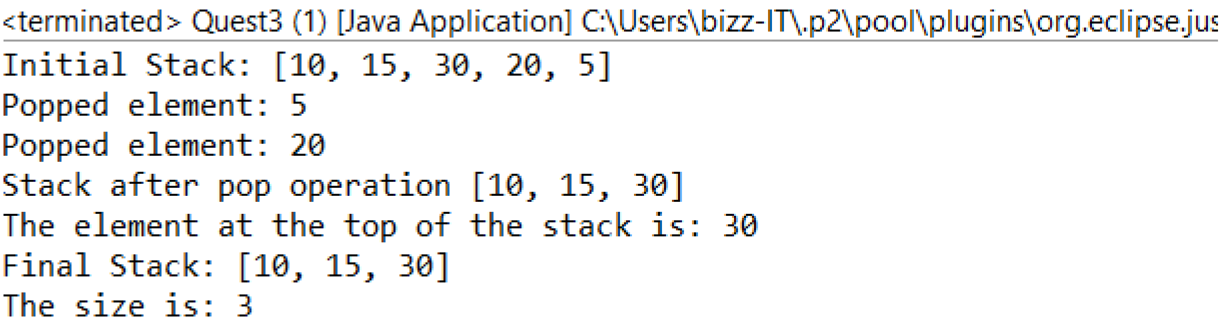
```
<terminated> Quest2 (1) [Java Application] C:\Users\bizz-IT\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86
10
12
20
100
155
Does the Queue contains 12? true
Priority queue after Insertion: [10, 12, 20, 100, 155, 1000, 2000]
top element of PriorityQueue: 10
top element and removing from the PriorityQueue container: 10
new top element: 12
After Remove - [20, 100, 1000, 2000, 155]
the size of queue: 5
The head of the element20
after clear method the pqueue is: []
```

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

```
package sba3;
import java.util.Stack;

public class Quest3 {


        public static void main(String[] args) {
            // Creating an empty Stack
            Stack<Integer> stk = new Stack<Integer>();
            // Use add() method to add elements
            stk.push(10);
            stk.push(15);
            stk.push(30);
            stk.push(20);
            stk.push(5);
            // Displaying the Stack
            System.out.println("Initial Stack: " + stk);
            // Removing elements using pop() method
            System.out.println("Popped element: "+ stk.pop());
```

```
                    System.out.println("Popped element: "+ stk.pop());
                    // Displaying the Stack after pop operation
                    System.out.println("Stack after pop operation "+ stk);
                    // Fetching the element at the head of the Stack
                    System.out.println("The element at the top of the"+ " stack is: "
+ stk.peek());

                    // Displaying the Stack after the Operation
                    System.out.println("Final Stack: " + stk);
                    // Displaying the size of stack
                    System.out.println("The size is: " + stk.size());
                    }
                    }
```

//Output

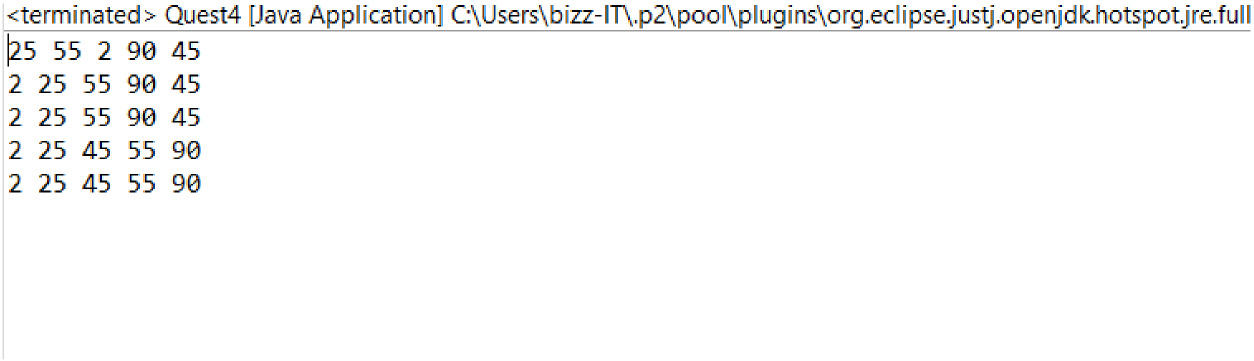


```
<terminated> Quest3 (1) [Java Application] C:\Users\bizz-IT\.p2\pool\plugins\org.eclipse.jus
Initial Stack: [10, 15, 30, 20, 5]
Popped element: 5
Popped element: 20
Stack after pop operation [10, 15, 30]
The element at the top of the stack is: 30
Final Stack: [10, 15, 30]
The size is: 3
```

4. Write a program to implement insertion sort

```
package sba3;
public class Quest4 {
public static void main(String[] args) {
                    int a[]= {25,55,2,90,45};
                    int temp,j; for(int i=1;i<a.length;i++)
                    { temp=a[i]; j=i;
                    while(j>0 && a[j-1]>temp)
                    {
                    a[j]=a[j-1];
                    j=j-1;
                    }
                    a[j]=temp;
                    for (int k=0; k<a.length; ++k)
                    {
                    System.out.print(a[k]+" ");
                    }
                    System.out.println();
                    }
                    for(int i=0;i<a.length;i++)
                    {
                    System.out.print(a[i]+ " ");
                    }
      }}
```

//Output



```
<terminated> Quest4 [Java Application] C:\Users\bizz-IT\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full
25 55 2 90 45
2 25 55 90 45
2 25 55 90 45
2 25 45 55 90
2 25 45 55 90
```