```java
/**
 * @(#)IntVect.java
 * IntVect class implemented with ArrayLists
 * Quiz 5' solutions are at the end of the class
 * @author Kadir Can Çelik
 * @version 1.00 2015/12/18
 */

import java.util.ArrayList;

public class IntVect {

        private ArrayList<Integer> nums;

    public IntVect() {
        nums = null;
    }

    public IntVect (ArrayList<Integer> nums) {
        this.nums = new ArrayList<Integer>();
        for ( int num : nums ) {
                this.nums.add (num);
        }
    }

    public String toString() {
        String result;
```

```
        result = "";

        if ( nums != null ) {

                for ( int i = 0; i < nums.size(); i++ ) {

                        result = result + i + ": " + nums.get (i) + "\n";

                }

        } else {

                result = "Null";

        }

        return result;

}


public void addElement (int element) {

    if ( nums == null ) {

            nums = new ArrayList<Integer>();

    }

    nums.add (element);

}


public void add (IntVect other) {

    if ( this.nums == null && other.nums == null ) {}

    else if ( this.nums == null && other.nums != null ) {

            this.nums = new ArrayList<Integer>();

            for ( int num : other.nums ) {

                    this.nums.add (num);

            }

    }

    else if ( this.nums != null && other.nums == null ) {}
```

```java
        else if ( this.nums.size() < other.nums.size() ) {

                int i;

                for ( i = 0; i < this.nums.size (); i++ ) {

                        this.nums.set (i, this.nums.get (i) + other.nums.get (i) );

                }

                for ( ; i < other.nums.size(); i++ ) {

                        this.nums.add ( other.nums.get (i) );

                }

        }

        else if ( this.nums.size() > other.nums.size() ) {

                for ( int i = 0; i < other.nums.size (); i++ ) {

                        this.nums.set (i, this.nums.get (i) + other.nums.get (i) );

                }

        }

        else {

                for ( int i = 0; i < this.nums.size(); i++ ) {

                        this.nums.set (i, this.nums.get (i) + other.nums.get (i) );

                }

        }

    }


    // !Quiz 5' Questions 2 and 4
```

2. Write the clone( ) method for IntVec but for the implementation assume that ArrayList is used.
4. Write the assign( ) method for IntVec but for the implementation assume that ArrayList is used.

```java
    // For question 4, just change the function signature as

    // public IntVect assign()

    public IntVect clone () {

        IntVect result;
```

```
        result = new IntVect();

        if ( nums != null ) {

                result.nums = new ArrayList<Integer>();

                for ( int num : nums ) {

                        result.nums.add (num);

                }

        }

        return result;

    }
```

1. To the IntVect class add another public method of type int with the name missingValue, an example for its use is myVec.missingValue(yourVec): this vector (myVec) and otherVector (yourVec) contains numbers from 1 to 100 -a particular value occurs only once- but one number of myVec is missing in yourVec, this method returns that value. The order of numbers in the vectors can be different. Assume that the implementation of IntVect is based on ArrayList.
For the source of the question see: http://javarevisited.blogspot.com.tr/2015/06/top-20-array-interview-questions-and-answers.html. The solution here is different from what is suggested in the link.

```
    // Since the question said that there are elements in either IntVects

    // No checking for null values were implemented

    public int missingValue (IntVect other) {

        int noOfOccur;

        int result;

        noOfOccur = 1;

        result = 0;

        for ( int i = 0; i < this.nums.size() && noOfOccur != 0; i++ ) {

                noOfOccur = 0;

                for ( int j = 0; j < other.nums.size(); j++ ) {

                        if ( this.nums.get (i) == other.nums.get(j) ) {

                                noOfOccur = noOfOccur + 1;
```

4

```
                    }

                }

                if ( noOfOccur == 0 ) {

                        result = this.nums.get (i);

                }

        }

        return result;

}



// Quiz 5' Question 3

    To the IntVect add another public method of type int with the name repeatValue, its typical use is
    myVec.repeatValue(yourVec): this vector (myVec) and otherVector (yourVec) contains numbers from 1 to
    100 -a particular value occurs only once- but one number of myVec appears twice in yourVec,  this method
    returns that value. The order of numbers in the vectors can be different. Assume that the implementation of
    IntVect is based on ArrayList. For the source of the question see:
    http://javarevisited.blogspot.com.tr/2015/06/top-20-array-interview-questions-and-answers.html. The
    solution here is different from what is suggested in the link.



// Since the question said that there are elements in either IntVects

// No checking for null values were implemented

public int repeatValue (IntVect other) {

        int noOfOccur;

        int result;

        noOfOccur = 0;

        result = 0;

        for ( int i = 0; i < this.nums.size() && noOfOccur != 2; i++ ) {

                noOfOccur = 0;

                for ( int j = 0; j < other.nums.size(); j++ ) {

                        if ( this.nums.get (i) == other.nums.get(j) ) {

                                noOfOccur = noOfOccur + 1;
```

5

```
                                    }

                        }

                        if ( noOfOccur == 2 ) {

                                    result = this.nums.get (i);

                        }

            }

            return result;

      }

}
```