## Problem 1

In the program below

```
public class Arr2DJagged {
    public static void main (String[] args) {
        int[][] arr =
            { {1,3}, {3,4,5,8}, {6,8}, {9} };
        double[] res = getAverages(arr);
        for (double e : res) System.out.print(e + " ");
    }
    // ...
}
```

add a static function **getAverages** which

- takes a two-dimensional array of **int**s (not necessarily rectangular);
- returns an array of numbers of type **double** with its size equal to number of 'rows' in the two-dimensional array passed to the function. Consecutive elements of the resulting array should be equal to arithmetic averages of all elements of consecutive rows of the input array.

For the array as above, the result should be

```
2.0 5.0 7.0 9.0
```

## Problem 2

Create a class containing static functions operating on arrays of type **int[]**:

- `static int[] add(int[] a, int elem)` — takes a sorted array `a` and creates and returns an array by one longer than `a` in which one element with value `elem` has been added. This new element should be added in such a way, that the resulting array is still sorted in nondecreasing order. Do not use sorting — the new element should be inserted into the resulting array directly on its correct location. The input array `a` may be of length 0.
- `static int[] delIndex(int[] a, int ind)` — takes an array `a` and creates and returns an array by one shorter than `a` in which there is no element which in `a` had index `ind`.
- `static int[] delFirst(int[] a, int e)` — takes an array `a` and creates and returns an array by one shorter than `a` in which there is no first element with value `e`. If there was no such element, the input array `a` is returned unmodified.
- `static int[] delLast(int[] a, int e)` — takes an array `a` and creates and returns an array by one shorter than `a` in which there is no the last element with value `e`. If there was no such element, the input array `a` is returned unmodified.

- `static int[] delAll(int[] a, int e)` — takes an array `a` and creates and returns a shorter array in which there is no elements with value `e`. If there was no such elements, the input array `a` is returned unmodified.
- `static void info(int[] a)` — takes an array `a` and prints it in one line with information about its size.

Remarks:

- Do not use any functions/classes from any packages except ***java.lang*** — in particular sorting or collections.
- Do not copy arrays or their subsequences using loops. Use instead the static method **arraycopy** from class **System** (from the ***java.lang*** package, no imports required)

  ```
  System.arraycopy(sArr, sIndex, tArr, tIndex, count)
  ```

  which copies count elements of the array sArr beginning at position sIndex to array tArr starting at position tIndex.
- In the **delIndex** function, the value of the index may be illegal — you don't have to handle such situation, just let the program crash.

For example, the following program

download *ArrAddRemove.java*

```
public class ArrAddRemove {
    public static void main(String[] args) {
        int[] a = {};
        info(a);          // line 1

        a = add(a, 4);
        a = add(a, 1);
        a = add(a, 3);
        a = add(a, 7);
        a = add(a, 4);
        a = add(a, 2);
        a = add(a, 7);
        a = add(a, 4);
        a = add(a, 8);
        a = add(a, 7);
        a = add(a, 4);
        a = add(a, 5);

        info(a);          // line 2

        a = delIndex(a, 2);
        a = delLast(a, 7);
        a = delFirst(a, 7);

        info(a);          // line 3
```

```
        a = delAll(a, 4);

        info(a);          // line 4
    }

    static int[] add(int[] a, int elem) {
        // ...
    }
    static int[] delIndex(int[] a, int ind) {
        // ...
    }
    static int[] delFirst(int[] a, int e) {
        // ...
    }
    static int[] delLast(int[] a, int e) {
        // ...
    }
    static int[] delAll(int[] a, int e) {
        // ...
    }
    static void info(int[] a) {
        // ...
    }
}
```

should print

```
Length 0: [ ]
Length 12: [ 1 2 3 4 4 4 4 5 7 7 7 8 ]
Length 9: [ 1 2 4 4 4 4 5 7 8 ]
Length 5: [ 1 2 5 7 8 ]
```

---