Problem 1

In the **main** function, create an array of **int**s of any size and content. The program prints, in one line, all elements of the array, but each value which appears in it only once.

For example, if the array is

```
int[] arr = {2, 5, 3, 2, 1, 3, 6, 5, 3};
```

the program should print

```
2 5 3 1 6
```

Do not create any additional arrays or collections. The array cannot be modified.

Problem 2

Write a program which rotates a given array of **int**s by one position to the left, i.e., the first element is replaced by the second, the second by the third etc., and the last by the first. For example, the array

```
1 2 3 4 5
```

after rotation should become

```
2 3 4 5 1
```

Do not create any auxiliary arrays or Strings!

After modifications of the size and/or values of elements of the array, the program should work correctly without any other changes.

Problem 3

Write a program which rotates a given array of **int**s by n position to the left, i.e., the first element should be replaced by $(n+1)$-st, the second by $(n+2)$-nd etc., and the first n elements should be placed at the end of the array. For example, the array

```
1 2 3 4 5
```

after rotation to the left by three position should become

```
4 5 1 2 3
```

Solve this problem in three ways:

- applying, in a loop, n times the rotation to the left by one position (see problem 2);
- creating one auxiliary array and copying elements of the arrays in loops;

- creating one auxiliary array and using the function System::arraycopy (without any explicit loops).

Note: Function arraycopy

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

copies count elements of the array sArr beginning at position sIndex to array tArr starting at position tIndex.

Problem 4

Write a program which rearranges the elements of an array of ints in such a way that all even elements appear before all odd elements. The program should print elements of the array before and after the operation and also the number of even elements.
For eample, for array

```
        int[] arr = { 2,3,4,3,6,7,6,8,2,9 };
```

the result could be (the order of printed values is irrelevant as long as even elements occur before odd ones):

```
        2 4 6 6 8 2 3 3 7 9 count = 6
```

Do not create any auxiliary arrays or Strings! Use only one pass of a loop!

Problem 5

Define an array of non-negative integers, and then print a vertical 'histogram' of data contained in the array, i.e., in consecutive columns as many asterisks as are the values of consecutive elements of the array; columns should be aligned at the bottom. For example, if the array is defined as

```
        int[] arr = {1, 5, 8, 2, 6};
```

the program should print

```
            *
            *
            *   *
          * *   *
          * *   *
          * *   *
          * * * *
        * * * * *
```

Problem 6

Write a program which defines *two* arrays of ints and prints all elements that occur in both arrays but each value once only, without repetitions.
For example for arrays

```
    int[] arr = { 2,3,4,3,6,7,6,8,2,9 };
    int[] brr = { 2,3,6,8,5,1 };
```

the result could be (the order of printed values is irrelevant):

```
    3 6 8 2
```

Do not create any auxiliary arrays, collections or **String**s. Do not use any classes from packages other than the standard ***java.lang***. Do not modify the arrays (in particular, do not sort them).

Problem 7

There are two arrays of **int**s, `a` and `b`. Elements of the first are in non-descending order, while those of the second in non-ascending order. Create an array `c` which can hold all elements from the two given arrays and copy elements from both of them to `c` in such a way, that its elements are in non-descending order.

For example, the following program

download *Merging.java*

```
import java.util.Arrays;

public class Merging {
    public static void main(String[] args) {
        int[] arra = {3, 5, 8, 9, 11, 13, 14, 14, 19};
        int[] arrb = {15, 11, 2, 2, 0, -1};

        int[] arrc = ...
        //
        // ...
        //

        System.out.println(Arrays.toString(arrc));
    }
}
```

should print

```
[-1, 0, 2, 2, 3, 5, 8, 9, 11, 11, 13, 14, 14, 15, 19]
```

After changing the lengths and/or values of arrays `a` and `b`, the program should work without any other modifications.