In each exercise make your source code and output readable.

Exercise 1. In a program, create an one-dimensional array of n random integers from the range [a, b], where n, a, b are given by the user. Compute and display how many positive numbers are in the array. List all positive numbers from the array. Some sample interaction with the program might look like this:

```
Give the length of the array: 10
Give the range [a,b]
a=-3
b=15
Numbers from the range [-3,15]:
-1   -2    2   -3    4    5   -3    6    7   -3
There are 5 positive numbers in the array:
2    4    5    6    7
```

Exercise 2. In a program, create an one-dimensional array of n random integers from the range [a, b], where n, a, b are given by the user. Compute and display the index of first occurrence of positive number in the array. Compute and display the index of last occurrence of positive number in the array. Some sample interaction with the program might look like this:

```
Give the length of the array: 10
Give the range [a,b]
a=-3
b=15
Numbers from the range [-3,15]:
-1   -2    2   -3    4    5   -3    6    7   -3
The first positive number is 2, and is located at index 2.
The last positive number is 7, and is located at index 8.
```

Exercise 3. In a program, create a one-dimensional array of n random integers from the range [a, b], where n, a, b are given by the user. Compute and display the index of first number that repeats in the array. Check whether in the array is at least one recurring number, and then output relevant information and if possible a number that recurs. Moreover, display the array where the number that recurs is surrounded by round brackets. Some sample interaction with the program might look like this:

```
Give the length of the array: 10
Give the range [a,b]
a=-3
b=4
Numbers from the range [-3,4]:
-1    -1    2   -3    2    3   -3    4   -1    4
Number -1 recurs, and the first occurrence is at index 0.
(-1)  (-1)    2   -3    2    3   -3    4   (-1)    4
```

Exercise 4. In a program create an array of n integers (n is given by the user), generate by the pseudorandom number generator elements of the array and store them in the array. Let generated number be from the range [0,10]. Display the array. Compute and display how many times each number from the range [0,10] occurs in the array. Some sample interaction with the program might look like this:

```
How many numbers? 30
3    6    7    1    8    3    1    0    6    1    1    6    1    8    4
8    3    2    3    0    4   10    1    5    4    6    2    4    4    2

 0: 2
 1: 6
 2: 3
 3: 4
 4: 5
 5: 1
 6: 4
 7: 1
 8: 3
 9: 0
10: 1
```