In each exercise make your source code and output readable.

Exercise 1. Write a program, which contains the following functions:
(a) A function "generate_array(a,n,p,q)" which generates pseudorandom number from the range [p,q] for the first n elements of the array a.
(b) A function "display_array(a,n)" which displays the values of the first n elements of the array on the screen.
(c) A function "count_occurrences(a,n,x)" which returns the number of occurrences of x in the array.
(d) A function "count_less(a,n,x)" which returns the number of values from the array less than x.
(e) A function "find_integer(a,n,x)" which returns the index of the first element from the array with value equal to x or -1 if the number x is not in the array.
(f*) A function "largest_less(a,n,x)" which returns the index of the first element from the array that has the highest value from elements with value less than x.
Test the functions in a suitably defined main program.
VERSION 1. In a program use the array of fixed size.
VERSION 2. In a program use dynamic array.

Exercise 2. Write a program that gets from the user one integer. The program should check and display the information whether the number is even or not, and the information whether the number is positive or negative. If zero is entered, only the message "zero" should be displayed on the screen. Sample output:

```
Enter integer: 3
3 is positive and odd

Enter integer: -5
-5 is negative and odd

Enter integer: 12
12 is positive and even

Enter integer: -6
-6 is negative and even

Enter integer: 0
zero
```

Extension 1. Extend the program by adding the possibility of repeating the calculation n times. The value of n is given by the user. Compute how many integers are odd.
Extension 2. Extend the program by adding the possibility of repeating the calculations as many times as the user wants. Ask the user after each calculation whether to repeat the calculations again. Compute arithmetic average of odd numbers.
Extension 3. Extend the program by adding the possibility of repeating the calculation at most n times, in that case zero stops repetitions. The value of n is given by the user. Find the integer with the highest value from integers given by the user.