

In each exercise make your source code and output readable.

The practical test will cover the following topics:

- **statements**
- **arrays**
- **functions**

All this listed above topics must appear in your practical test to get satisfactory mark, which is 3.

During practical test student is to:

- get exercises,
- write programs in C++ language solving problems stated in exercises,
- compile programs with options **-std=c++11 -Wall -pedantic-errors**,
- run and test written program.

The program written during a practical test cannot be identical to the programs of other students or very similar.

Exercise 1. Write a program, which contains the following functions:

- (a) A function "generate_array(a,n,p,q)" which generates random number from the range [p,q] for n elements of the array a.
- (b) A function "display_array(a,n)" which displays 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 "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.

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, which contains the following functions:

- (a) A function "generate_array(a,n,p,q)" which generates random number from the range [p,q] for n elements of the array a.
- (b) A function "display_array(a,n)" which displays n elements of the array on the screen.
- (c) A function "count_less(a,n,x)" which returns the number of values from the array less than x.
- (d*) 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 3. Write a program that gets from the user an array of integers of size 10.

Next, the program

- (a) computes and displays the arithmetic average of elements from the array.
- (b) computes and displays the arithmetic average of elements with values that belongs to the interval [-5,5].
- (c) computes and displays the arithmetic average of elements with values that belongs to the interval [p,q].
- (d) computes and displays the sum of all odd values from the array provided that at least one odd value occurs in the array
- (e) computes and displays the number of values from the array that are divisible by 3 provided that at least one divisible by 3 value occurs in the array
- (f) computes and displays the greatest value from the array.
- (g*) computes and displays the greatest odd value from the array.

VERSION 1. In a program use the array of fixed size.

VERSION 2. In a program define and call functions to tasks (a)-(f) as well as functions responsible for getting values from the user, outputting array, or generating the array by aid of pseudorandom number generator.

Exercise 4. 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 (Complex). 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.

Exercise 5. Write a program that gets from the user one value representing side an equilateral triangle. If the side is positive calculate the area and the circumference of the equilateral triangle given by the formulas: $A = \frac{\sqrt{3}}{4} a^2$, $C = 3a$, where a denotes the side of the equilateral triangle. Display all results obtained on the screen. In this program:

- write and call a function called **triangleArea** that computes the area of a equilateral triangle. It should take an argument of type float and return a value of the same type.

- write and call a function called **triangleCircumference** that computes the circumference of a equilateral triangle. It should take an argument of type float and return a value of the same type.

Extension 1. Extend the program by adding the possibility of repeating the calculation n times. The value of n is given by the user.

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.

Extension 3. When the calculations are repeating find the equilateral triangle with the smallest area.