

## Задания по «Функциональному программированию»

В этом задании нужна реализовать функцию, принимающую на вход массив(ы) вещественных данных и указатель на функцию, и применяющую к ним функцию определенным образом. Юнит-тесты обязательны.

*Замечание 1. Если говорится, что результат работы функции — массив, то это значит, что она должна принимать как параметр указатель на начало массива, в который будет осуществлять вывод.*

*Замечание 2. Предполагается, что все данные типа double.*

*Замечание 3. Массивы задаются в «итераторном» стиле, т.е. в виде двух границ. Если передается два массива одинаковой длины, то верхняя граница второго массива не указывается. Запрещается двигаться по итераторам непоследовательно, т.е. запрещены любые операции, кроме ++ и \*.*

1. `map2 (zip)`. Принимает на вход два массива одинаковой длины и двухместную функцию, возвращающую `double` (т.е.  $f: R \times R \rightarrow R$ ). Заполняет массив, такой же длины, как исходные, следующим образом:  $z_i = f(x_i, y_i)$ . Возвращает `void`.
2. `replaceIf (replace)`. Принимает на вход два массива одинаковой длины и предикат (т.е.  $f: R \rightarrow \{\text{true}, \text{false}\}$ ). Результат — первый массив, но в тех местах, где предикат истинен, значения заменяются на соответствующие значения второго массива. Возвращает количество измененных значений.
3. `replaceIf2 (replace2)`. Принимает на вход два массива одинаковой длины и двухместный предикат (т.е.  $f: R \times R \rightarrow \{\text{true}, \text{false}\}$ ). Результат — первый массив, но в тех местах, где предикат истинен, значения заменяются на соответствующие значения второго массива. Возвращает количество измененных значений.
4. `filter`. Принимает на вход массив и предикат (т.е.  $f: R \rightarrow \{\text{true}, \text{false}\}$ ). Заполняет массив только теми значениями исходного, для которых предикат верен. Возвращает количество «отфильтрованных» элементов.
5. `fold`. Принимает на вход массив и двухместную функцию, возвращающую `double` (т.е.  $f: R \times R \rightarrow R$ ). Заполняет массив на единицу меньшей длины следующим образом:  $z_1 = f(x_1, x_2)$ ,  $z_i = f(z_{i-1}, x_{i+1})$
6. `count (countIf)`. Принимает на вход массив и предикат (т.е.  $f: R \rightarrow \{\text{true}, \text{false}\}$ ). Возвращает количество элементов, на которых предикат верен.
7. `sort`. Принимает на вход массив и функцию сравнения элементов (двухместный предикат,  $\text{less}: R \times R \rightarrow \{\text{true}, \text{false}\}$ ), сортирует массив.

8. `apply2`. Принимает на вход массив и двухместную функцию, возвращающую `double`. Заполняет массив длины на 1 меньше исходного результатами применения функции к последовательным парам элементов исходного массива.
9. `apply3`. Принимает на вход массив и трехместную функцию, возвращающую `double`. Заполняет массив длины на 2 меньше исходного результатами применения функции к последовательным тройкам элементов исходного массива.