

# Контрольные вопросы:

1. (6 б.) В чем заключается концепция встраивания вызовов функций?
2. (6 б.) В каких случаях используется ключевое слово `auto`?
3. (6 б.) Какие аргументы функции могут иметь значения по умолчанию?
4. (6 б.) Что такое перегрузка функций?
5. (6 б.) Как передать функцию в функцию? Как сохранить лямбда-функцию?
6. (10 б.) Что такое ссылка? Чем отличается передача аргументов в функцию по ссылке и по значению? Как принято передавать массивы и другие контейнеры в функции? Может ли ссылка быть константной? Что это означает?

# Упражнения:

1. (25 б.) Реализуйте алгоритм вычисления заданного пользователем числа Фибоначчи, используя рекурсию.
2. (25 б.) Реализуйте алгоритм сортировки слиянием массива чисел. Потренируйтесь использовать рекурсию и не злоупотребляйте созданием вспомогательных массивов. Псевдокод алгоритма можете посмотреть [здесь](#).
3. (25 б.) Реализуйте алгоритм сортировки массива чисел по возрастанию и по убыванию, используя алгоритм `std::sort` из стандартной библиотеки и 2 лямбда-выражения для указания критериев сортировки массива. Посмотрите примеры в документации!
4. (25 б.) Реализуйте функцию `calculate`, которая принимает два числа с плавающей точкой и лямбда-выражение, и возвращает результат, вычисленный лямбда-выражением на основе переданных аргументов.

Например: `double result = calculate(1.23, 4.56, [](double x, double y) { return x + y; });` В результате выполнения данного фрагмента кода в `result` должно получиться значение 5.79. Для работы с лямбда-выражением используйте обертку на основе `std::function`. Создайте массив на основе контейнера `std::vector`, в котором будут храниться различные лямбда-выражения, и организуйте цикл, в котором к заданной паре чисел посредством функции `calculate` применяется каждое лямбда-выражение из массива.