

# Общие требования для всех лабораторных работ:

- Все задания реализуются на языке программирования C (стандарт C99 и выше).
- Реализованные в заданиях приложения не должны завершаться аварийно.
- Код должен компилироваться и корректно работать на всех операционных системах
- Во всех заданиях запрещено использование глобальных переменных (включая `errno`).
- Во всех заданиях запрещено использование оператора безусловного перехода (`goto`).
- Во всех заданиях запрещено пользоваться функциями, позволяющими завершить выполнение приложения из произвольной точки выполнения, вне контекста исполнения функции `main`.
- Запрещается использование VLA.
- Во всех заданиях при реализации необходимо разделять контексты работы с данными (поиск, сортировка, добавление/удаление, модификация и т. п.) и отправка данных в поток вывода / выгрузка данных из потока ввода.
- Определение и объявление функций должны быть разбиты на отдельные файлы (`.c`, `.h`)
- Во всех заданиях все параметры функций и вводимые (с консоли, файла, командной строки) пользователем данные должны подвергаться валидации в соответствии с типом валидируемых данных, если не сказано обратное; валидация должна зависеть от типа данных и логики применения этих данных для выполнения целевой подзадачи. При передаче аргументов приложению в командную строку, их количество также должно валидироваться.
- Во всех заданиях необходимо контролировать ситуации с невозможностью [пере]выделения памяти; во всех заданиях необходимо корректно освобождать всю выделенную динамическую память.
- Все ошибки, связанные с операциями открытия файла, должны быть обработаны; все открытые файлы должны быть закрыты.
- Не допускается утечка каких-либо ресурсов
- Во всех заданиях при реализации функций необходимо обеспечить возможность обработки ошибок различных типов на уровне вызывающего кода при помощи возврата целевых результатов функции через параметры функции и возврата из функции значения (либо типа `int`, либо перечислимого типа (`enum`)), репрезентирующего статус-код функции, в целях обработки последнего в вызывающем коде через оператор `switch/case` либо через управляющие конструкции языка `if / else if / else`. Возвращаемые статус-коды функций необходимо продумать самостоятельно так, чтобы были покрыты всевозможные ошибки времени выполнения функций.
- Во всех заданиях сравнение (на предмет эквивалентности или отношения порядка) вещественных чисел на уровне функции должно использовать значение эпсилон, которое является параметром этой функции.
- Во всех заданиях при реализации функций необходимо максимально ограничивать возможность модификации (если она не подразумевается) передаваемых в функцию параметров (используйте ключевое слово `const`).
- Для каждой задачи должен быть написан набор тестов (и программа соответственно должна их проходить)
- Во всех заданиях сравнение (на предмет эквивалентности или отношения порядка) вещественных чисел на уровне функции должно использовать значение эпсилон, которое является параметром этой функции