

# Задания к работе №3 по прикладному программированию.

Все задания выполняются на языке программирования C (стандарт C99 и выше).

Реализованные приложения не должны завершаться аварийно.

Все ошибки, связанные с операциями открытия файла, должны быть обработаны; все открытые файлы должны быть закрыты.

Во всех заданиях запрещено использование глобальных переменных.

Во всех заданиях сравнение (на предмет эквивалентности или отношения порядка) вещественных чисел на уровне функции должно использовать значение эпсилон, которое (предпочтительнее) является параметром этой функции, либо (на худой конец) является значением, определённым глобально.

Во всех заданиях при пользовательском вводе с консоли необходимо организовать приглашение пользователя ко вводу в соответствии с типом и бизнес-логикой запрашиваемых данных.

Для всех реализованных алгоритмов необходимо уметь объяснить их асимптотические сложности.

Во всех заданиях при реализации функций необходимо максимально ограничивать возможность модификации (если она не подразумевается) передаваемых в функцию параметров (используйте ключевое слово *const*).

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 через командную строку, их количество также должно валидироваться.

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.

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* в качестве точки входа (то есть вызванной не рекурсивно).

1. Реализовать и продемонстрировать работу функции, возвращающей последовательность (в виде динамического массива) из первых  $n$  (параметр функции) чисел Фибоначчи, используя рекурсию.
2. Описать тип структуры *Student* с полями: имя (*char \**), фамилия (*char \**), группа (*char \**), оценки по предметам (*int \**). Реализуйте функцию, на вход которой подаётся массив структур типа *Student* и значение *min\_avg* типа *double*. Функция должна вернуть односвязный список копий исходных структур с информацией о студенте, отсортированный по значению группы (по лексикографическому компаратору), чей средний балл не ниже *min\_avg*.
3. С консоли у пользователя запрашивают четыре значения  $a, b, c, d$ . При этом  $a, c$  - целые неотрицательные;  $b, d$  - целые отличные от 0. Используя считанные значения как числители и знаменатели дробей  $a/b$  и  $c/d$ , вывести в стандартный поток вывода сумму дробей, разность дробей, произведение дробей, частное дробей. Выведенные в стандартный поток вывода представления результатов операций должны быть несократимыми дробями.
4. Приложение загадывает псевдослучайное число от 1 до  $N$ . Пользователь вводит число. Программа сообщает: "EQ", "LT" или "GT" и так до тех пор, пока загаданное число не будет угадано пользователем. Подсчитать количество шагов до угадывания числа и вывести это значение в стандартный поток вывода. Предложите эффективную стратегию для угадывания числа.
5. С консоли у пользователя запрашивается значение целого неотрицательного числа из диапазона  $[0..10^9)$ . Для каждой из оснований систем счисления из диапазона  $[2..36]$  и значения считанного числа вывести в стандартный поток вывода гистограмму частотного распределения значений цифр числа. Пример: для числа  $12398_{10}$  и системы счисления с основанием 3 гистограмма может иметь следующий вид:  
  
0 \*\*\*\*\*  
  
1 \*\*  
  
2 \*\*\*
6. Реализовать приложение, выполняющее шифрование строки, переданной приложению как аргумент командной строки, при помощи шифра Цезаря. Для начала необходимо проверить, является ли строка такой, что состоит только из заглавных букв латинского алфавита. Если не является, необходимо сообщить пользователю об ошибке во входных данных. Если является, выполнить шифрование значением  $N$  (натуральное число в диапазоне  $[1..1000]$ ) и результат операции вывести в стандартный поток вывода.

7. Во входном файле, путь к которому является аргументом командной строки, построчно записаны значения результатов некоторых измерений длины в формате

*<значение длины> <единица измерения>*

. При этом каждый результат измерения содержит значение длины и строковое представление единицы измерения (“m” - метры, “cm” - сантиметры, “ya” - ярды, “ft” - футы, “lok” - локоть, “ver” - вершок, “arsh” - аршин, “sazh” - сажень). Необходимо перезаписать файл таким образом, чтобы его строки были отсортированы по значению длин с учётом их единиц измерения.

8. Реализовать приложение, получающее через аргументы командной строки значение натурального числа  $N$  и значение натурального числа  $base$  из диапазона  $[2..36]$  и выводящее в стандартный поток вывода
- Все простые числа в диапазоне  $[1..N]$ ;
  - Все натуральные числа в диапазоне  $[1..N]$ , сумма цифр которых в системе счисления с основанием  $10$  является простым числом;
  - Все натуральные числа в диапазоне  $[1..N]$ , произведение цифр представления в системе счисления с основанием  $base$  которых является простым числом;
  - Все натуральные числа в диапазоне  $[1..N]$ , произведение цифр представления которых в системе счисления с основанием  $base$  которых не является простым числом;
  - Все натуральные числа в диапазоне  $[1..N]$ , представления которых в системе счисления с основанием  $base$  является палиндромом (без ведущих нулей);
  - Все натуральные числа в диапазоне  $[1..N]$ , являющиеся простыми числами Софи Жермен (такие простые  $p$ , для которых  $2p+1$  также простые).
9. Реализуйте простую игру на угадывание "Быки и коровы". Программа должна хранить массив из четырёх попарно различных цифр в диапазоне от 0 до 9 (т.е., например, 2345, но не 4455), а пользователь должен угадать загаданное число. Допустим, программа загадала число 1234, а пользователь ввёл число 1359; программа должна ответить “1 бык и 1 корова”, поскольку пользователь угадал одну правильную цифру (1) в правильной позиции (бык) и одну правильную цифру (3) в неправильной позиции (корова). Угадывание продолжается до тех пор, пока пользователь не получит четырёх быков, т.е. не угадает четыре правильные цифры в четырех правильных позициях.