

Домашние задания на первый семестр

Общее описание домашних заданий

В первом семестре подразумевается выполнение трёх логически связанных домашних заданий. Конечной целью всех трёх домашних заданий является разработка ПО.

Для решения поставленной цели необходимо решить следующие задачи:

1. Подготовить среду функционирования и разработки ПО.
2. Подготовить систему сборки ПО.
3. Разработать архитектуру ПО.
4. Разработать ПО.

Первое домашнее задание решает задачу подготовки среды функционирования и разработки ПО, в связи с этим, подзадачами первого домашнего задания являются:

1. Установить систему контроля версий (Git Mercurial).
2. Установить среду разработки (VSCode, CodeBlocks, Eclipse, Clion, Visual Studio).

Во втором домашнем задании решаются вопросы подготовки системы сборки ПО и разработки архитектуры ПО. Таким образом, подзадачами второго домашнего задания являются:

1. Освоение сборки ПО с использованием стэке через командную строку. Для демонстрации успешного освоения использовать код пятой лабораторной работы.
2. Разработка архитектуры целевого ПО, включающая в себя:
 - согласование с заказчиком (преподавателем) окончательного условия задачи (ТЗ);
 - согласование форматов входных и выходных данных;
 - согласование ограничений на входные данные;
 - согласование с преподавателем разбиения программы на модули (различные .hpp и .cpp файлы) и назначение модулей;

- разработка и ручной расчёт нескольких тестовых примеров (достаточно трёх).

В третьем домашнем задании решается непосредственно задача разработки целевого ПО. В связи с этим подзадачами третьего домашнего задания являются:

1. Кодирование ПО с использованием установленной системы контроля версий (минимум 4 осмысленных commit'а).
2. Сборка ПО с использованием make.
3. Проверка работы ПО на тестовых примерах, рассчитанных во втором ДЗ.

Второе и третье домашнее задание подразумевают работу в команде. Команда состоит из трех студентов, совместно решающих один вариант.

Варианты домашнего задания

Вариант 1.

Разработайте приложение-архиватор, предназначенное для сжатия файлов, в которых значение каждого байта меньше 0x80. Сжатие производить следующим методом: если подряд встречается несколько одинаковых байт (например 0x70 0x70 0x70), то удалять все дубли, а вместо них добавлять их количество + 0x80 (то есть для приведенного случая должно получиться 0x83 0x70). Приложение должно уметь как архивировать, так и разархивировать файлы указанным методом.

Вариант 2.

Разработайте приложение для прохождения 2D лабиринтов. Лабиринт представляйте в виде двумерного массива, состоящего из 0 и 1. Для прохождения рекомендуется использовать волновой алгоритм.

Вариант 3.

Разработайте приложение-шифратор, которое принимает на вход: файл с открытым или зашифрованным текстом, файл с ключом и целое число N. Алгоритм шифрования заключается в следующем: ключ накладывается на текст посредством операции XOR, если остался необработанный текст, то ключ циклически сдвигается на N **бит** вправо и операция повторяется.

Вариант 4.

Разработайте ПО для хеширования файлов. Алгоритм хеширования является следующим:

1. Файл разбивается на части по 3 байта (если длина файла не кратна 3, то файл дополняется нулями).
2. Затем каждый три байта представляются в виде числа.
3. Если число получается чётным, то полученные три байта представляются в виде коэффициентов квадратного уравнения (a, b и c) соответственно, затем вычисляется дискриминант полученного выражения, и полученный дискриминант по модулю 0x100 записывается в выходной файл.

4. Если число получается нечётным, то записываемое в выходной файл значение считается по формуле $y = (a * b + c) \bmod 0 \times 100$
5. Если длина получившегося выходного файла > 2 байт, то полученный файл заново хешируется тем же алгоритмом.

Вариант 5.

Разработайте ПО для поиска одного корня степенных уравнений до 10-ой степени включительно. Для поиска корня используйте генетический алгоритм. Возможный источник информации о данном типе алгоритмов: <https://mathmod.asu.edu.ru/images/File/ebooks/GAfinal.pdf>.

Вариант 6.

Реализуйте ПО для предсказания ситуации в популяции вида по годам. Изначально в популяции две особи (одна мужская, другая женская), которым исполнился один год. Также известно, что есть запас N еды на начало расчётного периода, и каждый год удаётся закупить M еды.

Каждая особь в популяции характеризуется следующими признаками:

- пол;
- возраст;
- вероятность смерти (до 10 лет = 0, от 10 до 15 = const_die, после 15 = 100);
- вероятностью воспроизведения потомства (от 0 до 1 года = 0, от 1 до 13 – const_born; после 13 лет = 0);
- потребление пищи за год (от 0 до 1 года = 0, затем const_eat).

Для каждой особи в начале алгоритма расчёта текущей ситуации определяется, выжила она или нет в текущем году.

Затем для всех выживших особей определяется, хватит ли им на год еды. Если еды не хватит, то «убивается» случайная особь, пока количества еды не станет хватать на всех.

Затем для всех выживших производится «создание пар для воспроизведения потомства». Для каждой пары рассчитывается вероятность

воспроизведения потомства, как наименьшая из вероятностей воспроизведения потомства обоих родителей.

Затем производится «воспроизведение» потомства для каждой пары в соответствии с полученными ранее вероятностями (считается, что рождается один ребенок, пол которого является случайным).

Затем происходит переход на следующий год.

Вариант 7.

Разработайте ПО, предназначенное для анализа трафика пассажиропотока в метро в конкретный день недели. Всего анализируется трафик N пассажиров, про каждого из которых известно следующее:

- Для каждого дня недели начальная и конечная остановки следования пассажира (в какие-то дни пассажир может никуда не ездить).
- Номера поездов, на которых следовал пассажир. Каждый пассажир ездит только два раза, один раз в прямом направлении, а другой раз в обратном.

Определите, на каком перегоне и в каком поезде было больше всего пассажиров в каждый день недели. Считается, что один поезд в день едет один раз в прямом направлении и один раз в обратном.

Вариант 8.

Разработайте ПО для составления гардероба. Модник приходит в магазин, в котором есть M рубашек, N брюк, K пиджаков и L – пар обуви.

Рубашки, брюки и пиджаки характеризуются ценой и цветом, а каждая пара обуви характеризуется ценой и типом (кроссовки или ботинки).

Определите сколько различных вариантов комплектов одежды модник может подобрать в магазине, если:

1. У него есть D денег.
2. Комплект одежды должен состоять как минимум из рубашки, брюк и одной пары обуви.
3. Модник в силу своих идейных убеждений не включает в один комплект кроссовки и пиджак.

4. Также модник не может допустить, чтобы в комплекте были вещи трёх разных цветов.

Вариант 9.

Разработайте ПО для составления расписания пар для одной группы.

При следующих условиях:

1. У группы 15 преподавателей ведут по одной паре.
2. У каждого преподавателя есть время, в которое он может вести пары (свое для каждого дня недели).
3. Воскресенье – выходной.
4. Студенты учатся пять дней в неделю (не обязательно подряд, то есть среди недели может быть выходной, а суббота – рабочей).
5. В день у студентов не может быть больше 4 пар.
6. В день у студентов не может быть больше одного окна.

Вариант 10.

Разработайте ПО, решающее следующую задачу оптимизации.

Представим, что на какой-либо объект пробрался вор. На объекте хранится N ценных вещей, каждая из которых имеет свою стоимость и вес в килограммах. Определите, какие вещи вору стоит украсть, чтобы максимизировать ценность вынесенных вещей, если он может унести с собой всего V килограмм.