

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ «ДИНАМИЧЕСКИЕ СТРУКТУРЫ ДАННЫХ»

Задания по спискам 1–16

Задания на деревья 17–20

1	Используя структуру стека, вычислить выражение, содержащее две операции: поиск минимума (обозначается $m(<число1>, <число2>)$) и поиск максимума (обозначается $M(<число1>, <число2>)$). Операции могут быть вложенными, например, $M(15, m(16, 8))$ (в данном выражении ищем минимум из 16 и 8, а потом ищем максимум от результата $m(16, 8)$ и 15, ответ 15). В качестве аргументов могут использоваться только целые положительные числа. Строка, вводимая пользователем, корректна и не содержит пробелов.
2	<p>Игра в слова. Дан набор слов. Известно, что эти слова использовались для игры, в которой очередное слово должно заканчиваться той буквой, на которую закончилось предыдущее слово. С какого слова начинается игра, не имеет значения. Если слово оканчивается на «ь», то используется предыдущая буква. Последнее слово цепочки должно заканчиваться буквой первого слова цепочки. Входной файл содержит все слова в одной строке, слова записаны строчными буквами русского алфавита, между словами может быть один или более пробелов. Требуется выстроить данные слова в цепочку согласно правилам игры. Достаточно найти одно подходящее решение. Возможен вариант, что решений не существует.</p> <p>Пример 1:</p> <p>(файл входных данных) хмель мороз налог лимон запах дым лошадь холм гол молох (файл выходных данных) налог гол лошадь дым мороз запах холм молох хмель лимон</p> <p>Пример 2:</p> <p>(файл входных данных) аквариум сирень жатва заноза нож авось тормоз мост (файл выходных данных) авось сирень нож жатва аквариум мост тормоз заноза</p>
3	<p>Напечатать в порядке возрастания первых n натуральных чисел, в разложение которых на простые множители входят только числа 2, 3, 5.</p> <p><i>Идея решения:</i> введем три очереди x_2, x_3, x_5 в которых будем хранить элементы, которые соответственно в 2, 3, 5 раз больше напечатанных, но еще не напечатаны. Рассмотрим наименьший из ненапечатанных элементов: пусть это x. Тогда он делится нацело на одно из чисел 2, 3, 5; x находится в одной из очередей и является в ней первым элементом (меньшие его уже напечатаны, а элементы очередей не напечатаны). Напечатав x, нужно изъять его из очереди и добавить в очередь кратные ему элементы. Длины очередей не превосходят числа напечатанных элементов. Изначально в очередях хранится по одному числу.</p>

4	<p>Абстрактная вычислительная машина имеет один регистр и шесть инструкций:</p> <p>LD A Помещает операнд A в регистр ST A Помещает содержимое регистра в переменную с именем A AD A Прибавляет значение A к регистру. Результат остается в регистре. SB A Вычитает значение A из регистра. Результат остается в регистре. ML A Умножает содержимое регистра на значение переменной с именем A. Результат остается в регистре. DV A Делит содержимое регистра на значение A. Результат остается в регистре.</p> <p>Дано выражение в постфиксной форме, состоящее из однобуквенных операндов и операций «+», «-», «*», «/». Написать программу вывода инструкций вычислительной машины, необходимых для вычисления выражения. Результат вычисления должен оставаться в регистре, разрешается использовать обозначения T_n для временных переменных.</p> <p>Пример: для выражения ABC*+DE-/ перечень инструкций имеет вид</p> <p>LD B ML C ST T1 LD A AD T1 ST T1 LD D SB E ST T2 LD T1 DV T2</p> <p><i>Замечание:</i> Постфиксной формой записи выражения a b называется запись, в которой знак операции размещен за операндами a b.</p> <p>Примеры: a - b → a b - a * b + c → a b * c + a * (b + c) → a b c + * a + b / c / d * e → a b c / d / e * +</p>
---	--

5	<p>Числовое кольцо. В кольце записаны N чисел, составляющих по часовой стрелке 3 числа: два слагаемых и сумму. Требуется написать программу, которая по строке чисел, считая ее кольцом, находит какое-нибудь решение в виде $A+B=C$. Все цифры числа должны входить в числа только один раз и в порядке следования в кольце. Цифр в кольце не более 1000.</p> <p>Формат входных данных: Входной файл содержит 1 строку, в которой без пробелов перечислены цифры кольца.</p> <p>Формат выходных данных: В выходной файл выводится тождество в виде $\langle \text{слагаемое1} \rangle + \langle \text{слагаемое2} \rangle = \langle \text{сумма} \rangle$ без пробелов внутри или слово «No», если решения не существует.</p> <p>Пример 1:</p> <p>(входные данные) 01902021 (выходные данные) 190+20=210</p> <p>Пример 2:</p> <p>(входные данные) 111111 (выходные данные) No</p>
6	<p>Для заданного подмножества набора костей домино определить, можно ли их выложить в ряд, не нарушая правил. Если можно, то представить один любой вариант такого разложения. Например, для входных данных 31, 00, 13, получаем ответ: некорректные входные данные; для входных данных 02, 04, 42 ответ: можно, 04, 42, 20. Использовать двухсвязный список.</p>

7	<p>Игра в «пьяницу». В игре в «пьяницу» карточная колода раздается поровну двум игрокам. Далее они вскрывают по одной верхней карте, и тот, чья карта старше, забирает себе обе вскрытые карты, которые кладутся под низ его колоды. Тот, кто остается без карт – проигрывает. Для простоты будем считать, что все карты различны по номиналу, а также, что самая младшая карта побеждает самую старшую карту ("шестерка берет туза"). Игрок, который забирает себе карты, сначала кладет под низ своей колоды карту первого игрока, затем карту второго игрока (то есть карта второго игрока оказывается внизу колоды). Напишите программу, которая моделирует игру в пьяницу и определяет, кто выигрывает.</p> <p>В игре участвует 10 карт, имеющих значения от 0 до 9, большая карта побеждает меньшую, карта со значением 0 побеждает карту 9.</p> <p>Входные данные Программа получает на вход две строки: первая строка содержит 5 чисел, разделенных пробелами—номера карт первого игрока, вторая – аналогично 5 карт второго игрока. Карты перечислены сверху вниз, то есть каждая строка начинается с той карты, которая будет открыта первой.</p> <p>Выходные данные. Программа должна определить, кто выигрывает при данной раздаче, и вывести слово first или second, после чего вывести количество ходов, сделанных до выигрыша. Если на протяжении 10^6 ходов игра не заканчивается, программа должна вывести слово botva.</p> <p>Примеры входные данные 1 3 5 7 9 2 4 6 8 0 выходные данные second 5</p>
8	<p>Многочлен $P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ с целыми коэффициентами можно представить в виде списка, в котором первое информационное поле соответствует коэффициенту при неизвестном, а второе показателю степени этого же неизвестного. Написать процедуры для ввода и вывода многочлена из строкового формата в список. В качестве строкового формата может использоваться запись типа: $52y^{10} - 3y^8 + y$, где знак ^ означает возведение в степень. Реализовать процедуру для приведения подобных слагаемых в многочлене. С помощью описанных процедур ввести многочлен из файла, привести подобные члены и вывести результирующий многочлен в конец исходного файла, расположив его члены в порядке убывания степеней. Использовать однонаправленный линейный список.</p>

9	<p>Реализуйте структуру данных "стек". Напишите программу, содержащую описание стека и моделирующую работу стека, реализовав все указанные здесь методы. Программа считывает последовательность команд и в зависимости от команды выполняет ту или иную операцию. После выполнения каждой команды программа должна вывести одну строчку. Возможные команды для программы:</p> <p>push n - Добавить в стек число n (значение n задается после команды). Программа должна вывести ok.</p> <p>pop - Удалить из стека последний элемент. Программа должна вывести его значение.</p> <p>back - Программа должна вывести значение последнего элемента, не удаляя его из стека.</p> <p>size - Программа должна вывести количество элементов в стеке.</p> <p>clear - Программа должна очистить стек и вывести ok.</p> <p>exit - Программа должна вывести bye и завершить работу.</p> <p>Входные данные: команды управления стеком вводятся в описанном ранее формате по 1 на строке. Гарантируется, что набор входных команд удовлетворяет следующим требованиям: максимальное количество элементов в стеке в любой момент не превосходит 100, все команды pop и back корректны, то есть при их исполнении в стеке содержится хотя бы один элемент.</p> <p>Выходные данные: протокол работы со стеком, по 1 сообщению в строке.</p> <p>Примеры:</p> <pre> push 3 ok push 14 ok size 2 clear ok push 1 ok back 1 push 2 ok back 2 pop 2 size 1 pop 1 size 0 exit bye </pre>
---	--

10	<p>Очередь. Реализуйте структуру данных "очередь". Напишите программу, содержащую описание очереди и моделирующую работу очереди, реализовав все указанные здесь методы. Программа считывает последовательность команд и в зависимости от команды выполняет ту или иную операцию. После выполнения каждой команды программа должна вывести одну строку.</p> <p>Возможные команды для программы:</p> <p>push n Добавить в очередь число n (значение n задается после команды). Программа должна вывести ok.</p> <p>pop Удалить из очереди первый элемент. Программа должна вывести его значение.</p> <p>front Программа должна вывести значение первого элемента, не удаляя его из очереди.</p> <p>size Программа должна вывести количество элементов в очереди.</p> <p>clear Программа должна очистить очередь и вывести ok.</p> <p>exit Программа должна вывести bye и завершить работу.</p> <p>Перед исполнением операций front и pop программа должна проверять, содержится ли в очереди хотя бы один элемент. Если во входных данных встречается операция front или pop, и при этом очередь пуста, то программа должна вместо числового значения вывести строку error.</p> <p>Входные данные: вводятся команды управления очередью, по одной на строке Выходные данные: вывести протокол работы очереди, по одному сообщению на строке</p> <p>Пример 1:</p> <pre>push 1 ok front 1 exit bye</pre> <p>Пример 2:</p> <pre>size 0 push 1 ok size 1 push 2 ok size 2 push 3 ok size 3 exit bye</pre>
----	--

11	<p>Баржа. На барже располагается K грузовых отсеков. В каждый отсек можно поместить некоторое количество бочек с одним из 10 000 видов топлива. Причём извлечь бочку из отсека можно лишь в случае, если все бочки, помещённые в этот отсек после неё, уже были извлечены. Таким образом, в каждый момент времени в каждом непустом отсеке имеется ровно одна бочка, которую можно извлечь, не трогая остальных. Будем называть такие бочки крайними.</p> <p>Изначально баржа пуста. Затем она последовательно проплывает через N доков, причём в каждом доке на баржу либо погружается бочка с некоторым видом топлива в некоторый отсек, либо выгружается крайняя бочка из некоторого отсека. Однако, если указанный отсек пуст, либо если выгруженная бочка содержит не тот вид топлива, который ожидалось, следует зафиксировать ошибку. Если на баржу оказывается погружено более P бочек или если после прохождения всех доков она не стала пуста, следует также зафиксировать ошибку. От вас требуется либо указать максимальное количество бочек, которые одновременно пребывали на барже либо зафиксировать ошибку.</p> <p>Входные данные: в первой строке три целых числа N, K и P ($1 \leq N, K, P \leq 100\,000$). Далее следует N строк с описанием действия, выполняемого в очередном доке. Если в нём происходит погрузка, то строка имеет вид «+ $A\ B$», где A — номер отсека, в который помещается бочка, а B — номер вида топлива в ней. Если же док занимается разгрузкой, то строка имеет вид «- $A\ B$», где A — номер отсека, из которого извлекается бочка, а B — номер ожидаемого вида топлива.</p> <p>Выходные данные: вывести либо одно число, равное искомому максимуму в случае безошибочного прохождения баржей маршрута, либо вывести слово «Error» в противном случае. Примеры</p> <p>Пример:</p> <p>(входные данные)</p> <pre> 6 1 2 + 1 1 + 1 2 - 1 2 - 1 1 + 1 3 - 1 3 </pre> <p>(выходные данные)</p> <pre> 2 </pre>
----	--

12	<p>Шарики. В одной компьютерной игре игрок выставляет в линию шарики разных цветов. Когда образуется непрерывная цепочка из трех и более шариков одного цвета, она удаляется из линии. Все шарики при этом сдвигаются друг к другу, и ситуация может повториться. Напишите программу, которая по данной ситуации определяет, сколько шариков будет сейчас уничтожено. Естественно, непрерывных цепочек из трех и более одноцветных шаров в начальный момент может быть не более одной.</p> <p>Входные данные: даны количество шариков в цепочке (не более 10^5) и цвета шариков (от 0 до 9, каждому цвету соответствует свое целое число).</p> <p>Выходные данные: требуется вывести количество шариков, которое будет уничтожено.</p> <p>Примеры:</p> <p>(входные данные) 5 1 3 3 3 2</p> <p>(выходные данные) 3</p> <p>(входные данные) 10 3 3 2 1 1 1 2 2 3 3</p> <p>(выходные данные) 10</p>
----	---

13	<p>Контейнеры. На складе хранятся контейнеры с товарами N различных видов. Все контейнеры составлены в N стопок. В каждой стопке могут находиться контейнеры с товарами любых видов (стопка может быть изначально пустой).</p> <p>Автопогрузчик может взять верхний контейнер из любой стопки и поставить его сверху в любую стопку. Необходимо расставить все контейнеры с товаром первого вида в первую стопку, второго вида — во вторую стопку и т.д.</p> <p>Программа должна вывести последовательность действий автопогрузчика или сообщение о том, что задача решения не имеет.</p> <p>Входные данные: в первой строке задается одно натуральное число N, не превосходящее 500. В следующих N строках описаны стопки контейнеров: сначала записано число k_i — количество контейнеров в стопке, а затем k_i чисел — виды товара в контейнерах в данной стопке, снизу вверх. В каждой стопке вначале не более 500 контейнеров (в процессе переноса контейнеров это ограничение может быть нарушено).</p> <p>Выходные данные: выведите описание действий автопогрузчика: для каждого действия укажите два числа — из какой стопки брать контейнер и в какую стопку класть. (Обратите внимание, что минимизировать количество операций автопогрузчика не требуется.) Если задача не имеет решения, выдайте одно число 0. Если контейнеры изначально правильно размещены по стопкам, выводить ничего не надо.</p> <p><i>Комментарий к примеру:</i> изначально в первой стопке лежат четыре контейнера — снизу контейнер с товаром первого вида, над ним — с товаром второго вида, над ним — третьего, и сверху — еще один контейнер с товаром второго вида.</p> <p>Примеры:</p> <p>(входные данные)</p> <pre> 3 4 1 2 3 2 0 0 </pre> <p>(выходные данные)</p> <pre> 1 2 1 3 1 2 </pre>
----	--

14	<p>Реализуйте с помощью двусвязного циклического списка следующую игру: Ученики школы встают в круг. Один ученик пишет программу, генерирующую случайное целое число из промежутка $[-10;10]$. Если выпало положительное число, то отсчет ведется «по часовой стрелке», если отрицательное, то «против часовой». Ученик, на котором остановился счет, не выбывает из круга, а делает «доброе дело» и уровень его рейтинга увеличивается на 1. Следующий отсчет начинается с ученика, стоящего в круге рядом с тем, который только что делал «доброе дело», с правого (если отсчет ведется по часовой стрелке) или с левого (против часовой). После того, как игра всем надоела, нужно вывести список учеников в порядке невозрастания рейтинга. Если имеется несколько учеников с одинаковым рейтингом, то их нужно выводить в том порядке, в котором они поступили в список. Исходный список фамилий учеников для игры находится в текстовом файле. В начале игры рейтинг каждого ученика равен 0. Во время игры нужно выводить протокол: какое случайное число выпало, какой ученик делал доброе дело и рейтинг ученика после совершения доброго дела. Количество раундов игры вводится пользователем с клавиатуры.</p>
15	<p>Используя структуру стека подсчитать значение арифметического выражения, записанного в обратной польской записи (постфиксной записи) при условии, что используются знаки операций $+$, $-$, $*$, $/$ и операнды являются вещественными положительными числами. Во вводимой пользователем строке числа и знаки операций разделяются одним пробелом.</p> <p>Замечание: Постфиксной формой записи выражения $a\ b$ называется запись, в которой знак операции размещен за операндами $a\ b$.</p> <p>Примеры:</p> $a - b \rightarrow a\ b -$ $a * b + c \rightarrow a\ b * c +$ $a * (b + c) \rightarrow a\ b\ c + *$ $a + b / c / d * e \rightarrow a\ b\ c / d / e * +$
16	<p>«Хитрый купец» Два купца отправились торговать за море. Каждый из них повез по N одинаковых тюков с товаром. В пути корабль попал в шторм и дал течь. Чтобы корабль не затонул, капитан приказал поднять весь груз на палубу корабля, расставить его вдоль бортов по периметру и выбросить за борт половину груза. Груз выбрасывается по следующему правилу:</p> <ul style="list-style-type: none"> каждый тюк получает порядковый номер, начиная с тюка, стоящего на носу корабля; их нумерация осуществляется по часовой стрелке; номер первого выброшенного тюка соответствует текущему числу месяца M; через K тюков следующий тюк с товаром выбрасывается. Один из купцов прознал про это правило. Как необходимо расставить ему груз, чтобы ни один из его тюков не был выброшен? <p>Использовать циклический однонаправленный список.</p>

17	<p>«Почтальон» Мудрый тритон работает почтальоном в Лукоморье. В перечень его обязанностей входит доставка корреспонденции птицам, разместившим свои гнезда на ветвях дуба (постепенно вытеснив оттуда русалку). Птицы вьют свои гнезда там, где ветка дерева разделяется на несколько более тонких веток. Лукоморье является районом с развитой инфраструктурой, поэтому все разветвления на дубе заняты гнездами птиц. За время работы тритон не только запомнил, где кто живет, но и уяснил, что если его путь проходит через чье-либо гнездо, и его обитатели не получают при этом корреспонденции, ему приходится извиняться за доставленные неудобства. Естественно, интеллигентный тритон стремится как можно меньше беспокоить обитателей дуба, поэтому он заранее планирует свой маршрут так, чтобы приносить минимальное число извинений.</p> <p>Напишите программу, которая по описанию мест обитания птиц и списку корреспонденции определяет, сколько раз придется извиняться мудрому тритону.</p> <p>Каждое гнездо дуба обозначено уникальным номером, что соответствует номеру квартиры, который мы иногда указываем, отправляя письма родным и близким по обычной почте. Гнездо, расположенное на первой развилке ствола имеет номер один. Тритон может только ползти по веткам или стволу дуба, прыжки и перелеты полностью исключены. Никакие ветки не соприкасаются и не пересекаются.</p> <p>Формат входного файла: в первой строке входного файла находится число M – количество гнезд на дубе ($0 < M < 1000$). В следующих M строках находится информация о размещении гнезд и количестве писем для каждого из возможных адресатов. Первые два числа p_i и l_i – количество соседей гнезда с номером i и количество направленных по этому адресу писем соответственно. Далее в строке приводятся p_i чисел – номера гнезд, являющихся соседними с гнездом с номером i.</p> <p>Формат выходного файла: в единственной строке выходного файла должно содержаться единственное число – минимальное количество извинений, которые придется приносить мудрому тритону, для того чтобы доставить все письма.</p> <p>Пример:</p> <p>(входной файл)</p> <pre> 5 3 2 2 5 4 1 1 1 1 1 4 2 2 3 1 1 3 1 </pre> <p>(выходной файл)</p> <pre> 2 </pre>
----	---

18 Чиновники. Есть министерство из n чиновников. Каждый из них может иметь только одного непосредственного начальника и произвольное количество подчиненных. Однако в министерстве работает только один чиновник, не имеющий начальников, – главный чиновник. Для того чтобы предприниматель мог получить некоторую лицензию, необходимо получить подпись на документе главного чиновника. Каждый чиновник для выставления своей подписи требует подпись на документе хотя бы одного из своих непосредственных подчиненных и, может быть, некоторую взятку – известное число u_i .

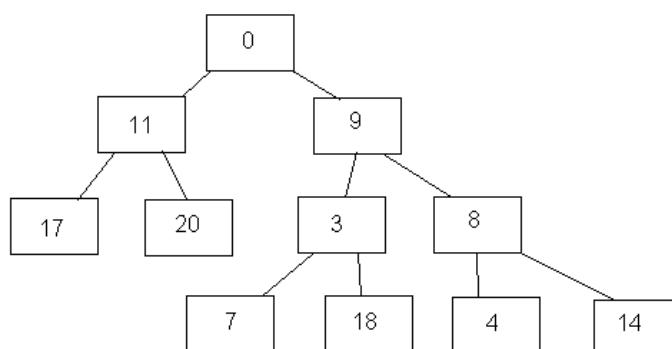
Необходимо определить, какое минимальное количество денег предпринимателю нужно заплатить, чтобы получить лицензию, и указать соответствующий этой сумме порядок получения подписей.

19 В файле даны n целых чисел, и здесь же указан путь их размещения в бинарном дереве виде двоичного кода (коды не повторяются). Построить двоичное дерево целых чисел, в котором путь по дереву определяется указанным двоичным кодом в этом листе (1 – переход к правому потомку, 0 – переход к левому потомку). В корень автоматически заносится значение 0.

Например, для исходных данных:

15	111
18	101
3	10
8	11
9	1
11	0
7	100
4	110
17	00
20	01

должно быть построено такое дерево:



Учитывать ситуацию, когда дерево не может быть построено.

20	<p>Написать программу, которая по заданной формуле строит дерево и производит вычисления с помощью построенного дерева. Формула задана в традиционной инфиксной записи, в ней могут быть скобки, максимальная степень вложенности которых ограничивается числом 10. Аргументами могут быть целые числа и переменные, задаваемые однобуквенными именами. Допустимые операции: +, -, *, /. Унарный минус допустим. С помощью построенного дерева формулы упростить формулу, заменяя в ней все поддеревья, соответствующие формулам $(f_1 * f_3 \pm f_2 * f_3)$ и $(f_1 * f_2 \pm f_1 * f_3)$ на поддеревья, соответствующие формулам $((f_1 \pm f_2) * f_3)$ и $(f_1 * (f_2 \pm f_3))$.</p>
----	---