

Задача А. Хладнокровный дуб

Имя входного файла:	tree.in
Имя выходного файла:	tree.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайта

Вампиры — странные существа. Они готовятся к проведению ежегодного праздника под названием «Бессмертие».

Для проведения праздника Эдварду было поручено подвесить за корень символ праздника — «Хладнокровный дуб».

«Хладнокровный дуб» является обычным деревом: в нем n вершин и $n - 1$ ребро, причем между любой парой вершин существует единственный простой путь, и корневая вершина имеет номер 1.

Из-за особенностей вампиров при нахождении под солнцем, главная часть праздника проводится ночью. Поэтому было решено использовать дуб в качестве освещения. Для этого к каждой некорневой вершине дуба можно подвесить сколько угодно лампочек. Для стабильности системы, к каждой лампочке проводят отдельную электрическую проводку из корневой вершины. Этот провод идет по пути дерева из корневой вершины в вершину с соответствующей ей лампочкой, проходя через все промежуточные вершины и ребра на пути.

Эдвард никак не может решить, в какие вершины и сколько лампочек нужно повесить.

Вам задана структура «Хладнокровного дуба», для каждой вершины $v > 1$ известна вершина p_v , за которую она подвешена. От вас требуется отвечать на странные вопросы Эдварда:

1. **add** x y — Эдвард вешает y лампочек в вершину с номером x ($2 \leq x \leq n$, $1 \leq y \leq 10^4$)
2. **del** x y — Эдвард убирает y лампочек из вершины с номером x ($2 \leq x \leq n$)
3. **ask** x — Эдвард просит посчитать $f(x)$ ($2 \leq x \leq n$)

$f(x)$ — это очень сложная функция, Эдварду не так просто ее описать, а о том, чтобы ее самому посчитать, он даже не задумывается. Вычисляется она так:

$$f(x) = \sum_{v \in ST(x)} g(v)$$

$ST(x)$ — это множество таких вершин, что, если в них повесить лампочку, то провод, который к ней придется провести, будет проходить через ребро $(x, p(x))$.

А $g(v)$ — это число проводов, в данный момент проходящих через ребро $(v, p(v))$.

Формат входного файла

В первой строке задано натуральное число n ($3 \leq n \leq 200\,000$) — число вершин в «Хладнокровном дубе».

Во второй строке записано $n - 1$ чисел: $p_2, p_3 \dots p_n$ ($1 \leq p_i \leq n$, $p_i \neq i$).

Гарантируется, что заданный «Хладнокровный дуб» является деревом.

В следующей строке задано натуральное число m ($1 \leq m \leq 200\,000$) — число вопросов Эдварда.

В следующих m строках заданы сами вопросы в формате, описанном в условии задачи.

Гарантируется, что все запросы корректны и Эдвард не будет убирать из вершины несуществующие лампочки.

Формат выходного файла

Для каждого вопроса Эдварда вида **ask** x выведите одно целое число — ответ на этот вопрос. Ответы требуется выводить в таком же порядке, в каком были заданы соответствующие им вопросы.

Примеры

tree.in	tree.out
2 1 4 add 2 1 ask 2 del 2 1 ask 2	1 0
5 1 2 2 1 12 add 3 4 ask 2 add 4 5 ask 2 del 3 1 ask 2 del 4 3 add 5 1 ask 5 del 5 1 ask 5 ask 2	8 18 16 1 0 10

Задача В. Запасы на зиму

Имя входного файла: `stock.in`
Имя выходного файла: `stock.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Всем известно, что вампиры пьют кровь. Большинство вампиров пьют кровь людей, но Эдвард не такой. Он не хочет убивать людей, поэтому он пьет только кровь животных и ту, которую люди сдают в пункты сдачи крови. Но скоро зима, животных будет не найти, а все пункты сдачи крови закроются. Поэтому Эдвард решил запастись кровью на зиму.

Эдвард знает n пунктов, в которых он сможет достать кровь. Одна проблема — пункты работают только в определенные моменты времени. Пункт номер i работает только с l_i минуты по r_i . Эдвард получит кровь в этом пункте только в том случае, если пробудет в нем все время работы, то есть в минуты с номерами с l_i по r_i .

Эдвард хочет достать как можно больше крови, чтобы зимой у него не было проблем. Для этого ему надо посетить некоторые пункты сдачи крови. Понятно, что он не может оказаться в двух местах в один и тот же момент времени, поэтому все пункты ему не всегда удастся посетить, но ему бы хотелось посетить как можно больше. Эдвард — вампир, поэтому он перемещается очень быстро, можно считать, что между пунктами сдачи крови он перемещается мгновенно. Помогите ему — скажите, какое максимальное количество пунктов сдачи крови ему удастся посетить.

Во всех пунктах Эдвард получает одинаковое количество крови.

Формат входного файла

В первой строке входного файла дано число n ($1 \leq n \leq 10^5$) — количество пунктов сдачи крови. В каждой из следующих n строк записано два числа l_i и r_i ($1 \leq l_i < r_i \leq 10^9$) — моменты времени, в которые работает пункт сдачи крови номер i .

Формат выходного файла

В первой строке выходного файла выведите максимальное количество пунктов сдачи крови, которые сможет посетить Эдвард. Во второй строке выведите номера этих пунктов. Номера выводите в том порядке, в котором Эдвард будет их посещать. Пункты нумеруются с 1 в порядке, в котором они заданы во входном файле.

Если ответов несколько, разрешается вывести любой.

Пример

<code>stock.in</code>	<code>stock.out</code>
2	2
1 2	1 2
3 4	
4	2
1 4	1 4
2 5	
3 6	
4 7	

Комментарий

В первом тестовом примере Эдварду ничего не мешает сначала посетить первый пункт сдачи крови, а затем второй.

Во втором примере он может сначала побывать в первом пункте в моменты времени с 1 до 4 и сразу же, в момент времени 4, оказаться в четвертом пункте.

Задача С. Нападение

Имя входного файла:	<code>war.in</code>
Имя выходного файла:	<code>war.out</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Всем известно о существовании вампиров. Однако, никто не задумывается о непосредственной близости этих существ. В сумеречном королевстве имеется n городов. Между некоторыми городами есть двусторонние дороги. В каждом городе есть свой вампирский клан, в который входит k_i вампиров.

В свое время вампиры изгнали из королевства всех оборотней, сделав их тем самым своими лютыми врагами. Оборотням это, естественно, не понравилось. Они решили объединиться в один отряд и напасть на какой-нибудь город. Отряд, нападающий на город, содержит w оборотней.

Однако, приняв в расчет то, что вампиры всегда действуют сообща, оборотни начали сомневаться в успешности их нападения. Теперь они пришли к вам за помощью. Известно, что вампиры могут перемещаться между двумя городами, соединенными дорогой, за один день. Также известно, что за один день сражения погибает $\min(t, a, b)$ вампиров, обороняющих осажденный город, и столько же оборотней, где a — текущая численность отряда вампиров, b — текущая численность отряда оборотней, t — константа. По данной вам информации, требуется узнать, смогут ли вампиры защитить город. Город считается осажденным, если существует момент времени, когда число оборотней, напавших на город, больше нуля, а число вампиров, обороняющих город, равно нулю.

Формат входного файла

В самой первой строке написано три числа: n ($1 \leq n \leq 10^5$) — количество городов, m ($1 \leq m \leq 10^5$) — количество дорог, t ($1 \leq t \leq 10^3$) — количество вампиров и оборотней, погибающих за один день сражения. Во второй строке написано ровно n чисел: k_i ($1 \leq k_i \leq 10^4$) — количество вампиров в i -м городе. Следующие m строк описывают дороги между городами: в каждой строке написано два числа — номера городов, соединенных дорогой. В последней строке написан номер города, на который было произведено нападение, и w ($1 \leq w \leq 10^4$) — размер армии оборотней.

Формат выходного файла

Выведите «Vampires win», если вампиры смогут отстоять свой город, и «Werewolves win» иначе.

Пример

war.in	war.out
5 6 5 4 5 4 4 1 1 5 1 3 2 1 3 5 2 5 4 3 1 8	Werewolves win
5 5 1 3 1 5 1 1 1 2 1 5 1 4 4 5 2 3 1 11	Vampires win

Комментарий

Обратите внимание, что в случае, если в городе погибают все вампиры, а новый отряд из соседнего города приходит на следующий день, вампиры все равно проигрывают.

Задача D. Лучшие друзья Беллы

Имя входного файла: diamonds.in
Имя выходного файла: diamonds.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайта

Скоро день рождения Беллы, и Эдвард собирается преподнести ей в подарок роскошное ожерелье из драгоценных камней. Ожерелье состоит из драгоценных камней нескольких типов. Но вот незадача: Эдвард не знает предпочтения Беллы и может только догадываться, какие камни ей нравятся.

Ювелир уже изготовил одно большое ожерелье, но он может вырезать из него некоторое количество подряд идущих камней и сделать из него новое ожерелье. Однако, есть еще одна проблема: Белле нравятся только те ожерелья, в которых количества камней каждого типа, которые ей нравятся, одинаковы. Количество тех камней, которые ей не нравятся, ей безразлично.

Драгоценные камни обозначаются строчными буквами английского алфавита. Вампиры крайне суеверны, и, в частности, камни, которым соответствуют гласные буквы, считаются приносящими беду, поэтому их в изначальном ожерелье нет. Согласными буквами являются все буквы английского алфавита, кроме букв A, E, I, O, U.

Эдвард пришел в ювелирный магазин, и у него есть q предположений по поводу предпочтений Беллы. Для каждого из этих предположений Эдвард хочет узнать, какова максимально возможная длина ожерелья, которую ювелир может получить из уже изготовленного. Новое ожерелье должно состоять из каких-то подряд идущих камней из уже готового ожерелья. Помогите ему!

Формат входного файла

В первой строке находится целое число n ($1 \leq n \leq 5000$) — длина уже изготовленного ожерелья. Во второй строке находятся n строчных согласных букв английского алфавита — описание этого ожерелья. В следующей строке находится целое число q — число предположений Эдварда ($1 \leq q \leq 10^5$). В каждой из следующих q строк находится описание очередного предположения: число k ($1 \leq k \leq 21$) — количество камней, которые, как кажется Эдварду, нравятся Белле. Через пробел идет строка из k строчных согласных букв английского алфавита —перечисление камней, которые могут нравиться Белле. Все символы в этой строке различны.

Формат выходного файла

Для каждого предположения Эдварда выведите ответ в отдельной строке. Ответом являются два числа l и r ($1 \leq l \leq r \leq n$), означающие, что Эдварду надо выбрать камни с l по r включительно. Количество букв, перечисленных Эдвардом в соответствующем предположении, должны совпадать, а длина должна быть максимальной. Если таких l и r не существует, выведите единственное число -1.

Примеры

diamonds.in	diamonds.out
6	1 5
bbcddb	-1
3	2 4
2 bd	
4 bcdf	
3 bcd	

Комментарий

Обратите внимание, что ожерелье незамкнуто, то есть оно представляет из себя просто набор подряд идущих драгоценных камней.

Задача Е. Чтение

Имя входного файла: `reading.in`
Имя выходного файла: `reading.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В необычной и насыщенной событиями жизни Беллы иногда находится место самым обыденным занятиям. Например, иногда она читает абсолютно обыкновенные книги. Однако, в последнее время Белле начали сниться странные сны, связанные с этим занятием.

В своих снах Белла также читала книгу, но текст в этой книге был не обычным художественным текстом, а случайным бредом, не несущим никакой смысловой нагрузки. При этом, сны были настолько реалистичными, что Белла стала периодически терять ощущение того, где она находится — во сне или наяву.

Помогите Белле по тексту, который напечатан в книге, которую она читает в данный момент, определить, находится ли она во сне или наяву.

Формат входного файла

Входной файл содержит некоторое количество символов, ASCII-код которых не превышает 127 — текст, напечатанный в книге, которую в данный момент читает Белла. Гарантируется, что количество символов в каждом тесте не превышает 50 000. Кроме этого, гарантируется, что нет ни одного теста, количество символов в котором меньше 10 000.

Известно, что этот текст является или отрывком некоторого художественного произведения на английском языке (возможно, переведенным с русского языка), или сгенерирован программой-генератором случайного текста. Код программы-генератора, написанный на языке Java, вы можете найти, перейдя по ссылке <http://neerc.ifmo.ru/school/io/reading/>

Формат выходного файла

Выведите в выходной файл фразу «Real life», если данный вам текст является художественным текстом, или же «Dreaming», если данный текст сгенерирован программой-генератором.

Пример

<code>reading.in</code>	<code>reading.out</code>
Having thanked Anna Pavlovna for her charming...	Real life
Lnх. Wlqytv gmudkiup fhtem, wyhprlrd...	Dreaming

Комментарий

По понятной причине, привести тест полностью в рамках условия не представляется возможным. Первые два теста доступны по ссылкам <http://neerc.ifmo.ru/school/io/reading/01.in> и <http://neerc.ifmo.ru/school/io/reading/02.in>.

В тестах, являющихся настоящими английскими текстами, могут встречаться опечатки и нарушения прафил орфографии и пунктуации, принятых в английском языке. Однако, жюри гарантирует, что они встречаются в количестве, не оказывающем существенного влияния на возможные решения этой задачи.

Задача F. Нападения

Имя входного файла: `attacks.in`
Имя выходного файла: `attacks.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайта

К сожалению, не все вампиры являются вегетарианцами, из-за чего периодически происходят неприятные события. Например, в последнее время в некоторых городах стали пропадать люди.

Чтобы добраться из одного города в другой вампиры могут пользоваться двусторонними дорогами. Про каждую дорогу известно, какие два города она соединяет, а также время в часах, которое необходимо вампирам, чтобы добраться из одного города в другой по этой дороге.

Вам известно, что всего было зафиксировано m фактов пропаж людей. Про каждое такое событие вам известно время, в которое оно произошло (в часах от начала месяца), а также город, в котором оно произошло. Необходимо определить минимальное количество вампиров, которые могут стоять за этим.

Нападения, совершенные в моменты времени t_1 и t_2 могут быть совершены одним и тем же вампиром, если он может добраться из города, в котором произошло одно событие, до города, в котором произошло второе, менее чем за $|t_2 - t_1|$ часов.

Найдите минимальное количество вампиров, которые могут стоять за атаками на людей.

Формат входного файла

В первой строке задано целое число T ($1 \leq T \leq 100$) — количество тестов. Каждый из тестов описывается следующим образом.

В первой строке задано три целых числа n , m и k ($1 \leq n, m \leq 500$, $1 \leq k \leq 10^5$) — количество городов, количество атак, а также количество дорог между городами. В следующих m строках записано по два числа t_i и v_i ($1 \leq t_i \leq 10^6$, $1 \leq v_i \leq n$) — время в часах, а также город, в котором произошла очередная атака. В следующих k строках содержится по три числа a_i , b_i и c_i ($1 \leq a_i, b_i \leq n$, $1 \leq c_i \leq 10^6$), которые описывают дорогу между городами a_i и b_i . Вампиры могут воспользоваться этой дорогой и добраться из одного города в другой потратив c_i часов.

Гарантируется, что суммарное количество городов во всех тестах не превышает 500. Аналогично, суммарное количество атак не превышает 500, а дорог — 100 000.

Формат выходного файла

Для каждого теста в отдельной строке выведите минимальное количество вампиров, которые могут стоять за нападениями на людей.

Примеры

<code>attacks.in</code>	<code>attacks.out</code>
2	2
2 3 1	1
10 1	
18 2	
20 1	
1 2 5	
2 3 2	
10 1	
18 2	
20 1	
1 2 5	
1 2 1	

Задача G. Соревнование по программированию

Имя входного файла: `contest.in`
Имя выходного файла: `contest.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайта

Чтобы Эдвард меньше охотился на невинных существ, Белла решила найти ему какое-нибудь увлекательное занятие. Всем известно, что нет ничего более увлекательного, чем участвовать в командных соревнованиях по программированию. Оказалось, что для участия в соревновании нужна команда из N человек, поэтому Эдвард позвал своих знакомых вампиров, а Белла — школьниц. Таким образом, в команде оказалось ровно N участников.

Соревнование идет ровно T минут. В отличие от обычных соревнований, каждому участнику полагается компьютер, на котором он может работать независимо от сокомандников. Командам предложено решить M задач. Про каждую из задач Эдварду известно, какие члены его команды могут ее решить. Так же ему известно, что на решение любой задачи любому члену команды, который умеет ее решать, потребуется ровно L минут, чтобы ее решить.

Штраф за задачу — время, прошедшее от начала соревнования, до того момента, как ее решили. Штраф команды — сумма штрафов за все задачи, которые она решила. Нерешенные задачи никак не влияют на штраф.

Эдварду интересно, какое максимальное количество задач сможет решить его команда при оптимальном распределении заданий между членами команды. Если существует несколько способов это сделать, Эдвард хотел бы минимизировать штраф.

Формат входного файла

Первая строка входного файла содержит числа N , M , T и L ($1 \leq N, M \leq 100$, $1 \leq L \leq T \leq 10000$) — количество членов команды, количество задач, продолжительность соревнования в минутах и время в минутах, за которое один участник способен решить одну задачу соответственно.

Следующие N строк содержат описание каждого участника: сначала идет число K ($0 \leq K \leq M$) — количество задач, которые умеет решать участник, затем K различных чисел — номера этих задач. Задачи нумеруются натуральными числами начиная с единицы.

Формат выходного файла

Выведите два числа — максимальное количество задач, которые успеет решить команда Эдварда и Беллы, и минимальный штраф, который при этом может быть достигнут.

Примеры

<code>contest.in</code>	<code>contest.out</code>
3 4 10 2 1 1 2 1 2 2 3 4	4 10

Комментарий

Первый участник решает первую задачу, второй вторую, а третий — последние две. Таким образом, суммарный штраф — $2+2+2+4=10$.

Задача Н. Ликантропия

Имя входного файла:	<code>stdin</code>
Имя выходного файла:	<code>stdout</code>
Ограничение по времени:	5 секунд
Ограничение по памяти:	256 мегабайт

Это интерактивная задача.

Джейкоб — оборотень, умеющий превращаться в волка. Сейчас он в теле человека и хочет превратиться в волка. Но его волчьей сущности надоели частые превращения туда-обратно и она пытается сопротивляться превращению.

Для превращения в волка, мозг Джейкоба посылает сигнал в нервную систему. Сигнал распространяется по нервной системе вниз, от мозга к сердцу. Нервная система Джейкоба состоит из n нейронов и m синапсов. Мозг генерирует сигнал в первом нейроне, после чего начинается его распространение до сердца (нейрона с номером n). Каждый синапс может передавать сигнал из a_i -го нейрона в b_i -й ($a_i < b_i$). После того, как сигнал приходит в нейрон, он должен отправиться дальше по какому-нибудь синапсу. Если синапса, по которому сигналу можно отправиться, нет, сигнал затухает и превращения в волка не происходит. Если сигнал доходит до сердца, начинается превращение в волка.

Волчья сущность хочет помешать Джейкобу начать превращение в волка. Для этого она, каждый раз, сразу после того, как сигнал приходит в нейрон, может временно дестабилизировать работу не более, чем k синапсов, и они не смогут переносить сигнал, пока он не уйдёт из того нейрона, в котором он находится в данный момент. После ухода сигнала, волчья сущность снова сможет дестабилизировать (возможно, другие) синапсы, и так далее, до затухания сигнала, либо до начала превращения в волка. Работоспособность ранее дестабилизированных синапсов при этом восстанавливается.

Помогите волчьей сущности провалить попытку превращения.

Протокол взаимодействия с программой жюри:

В первой строке заданы числа n , m и k ($2 \leq n \leq 50$, $0 \leq m \leq 10\,000$, $0 \leq k \leq m$) — число нейронов, число синапсов и максимальное число дестабилизируемых волчьей сущностью синапсов.

Далее, в m строках заданы описывающие синапсы числа a_i и b_i ($1 \leq a_i < b_i \leq n$) — откуда и куда может транспортироваться i -м синапсом сигнал.

Если не существует способа помешать превращению в волка, дестабилизируя при каждом перемещении сигнала не более k синапсов, чтобы не дать сигналу дойти до сердца, выведите «NO» и завершите программу.

Иначе, много раз повторяется следующее:

- Выведите число l ($0 \leq l \leq k$) — количество синапсов, работу которых необходимо дестабилизировать на этом этапе, и затем l различных чисел s_i ($1 \leq s_i \leq m$) — номера дестабилизируемых синапсов.
- Считайте очередное v — новое местоположение сигнала. Если $v = -1$, то завершите работу программы. Если сигнал дошёл до нейрона n , то волчья сущность проиграла.

Пример

stdin	stdout
4 4 2 1 2 1 3 2 4 3 4 -1	2 1 2
4 4 2 1 2 1 3 2 4 3 4 2 -1	0 1 3
2 3 2 1 2 1 2 1 2	NO

Комментарий

Для корректной работы программы после каждой операции вывода данных вам необходимо делать следующие операции:

- В языке Pascal: `flush(output);`
- В C/C++: `fflush(stdout);`
- В Java: `System.out.flush();`
- В Python: `sys.stdout.flush();`

Кроме этого, не забывайте после каждой выведенной строки ставить перевод строки.

Задача I. Мощь вампира

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайта

Джейкоб думает очень громко.

Эдвард

Это интерактивная задача.

Все вампиры обладают уникальной способностью читать мысли, и Эдвард очень часто пользовался ей для защиты своей жены Беллы Свон. Поэтому, когда на нее напали несколько вампиров клана Вольтури, он тут же решил прочитать их мысли, дабы узнать мощь каждого напавшего вампира.

Но каждый вампир этого клана обладает ментальным щитом, поэтому узнать мощь отдельного вампира невозможно. Эдвард может лишь узнать сумму всех чисел, выражающих их мощь, но при этом он может брать каждое число с нужным ему знаком (плюс или минус).

Эдварду срочно необходима ваша помощь, иначе Беллу съедят вампиры из клана Вольтури.

Протокол взаимодействия с программой жюри:

Программа жюри выводит число n в отдельной строке ($1 \leq n \leq 100$). После этого не более чем n раз повторяются следующие действия.

Ваша программа выводит в отдельной строке строку, состоящую из символов a_i ($a_i = +$ или $a_i = -$), разделенных пробелами, где i -й символ означает, с каким знаком мы просуммируем мощь i -го вампира p_i ($-100 \leq p_i \leq 100$).

Когда вы найдете мощь каждого вампира, вашей программе необходимо вывести в строчку их все по порядку:

`answer: a1 a2 ... an`

где a_i — мощь i -го вампира.

Примеры

stdin	stdout
2	+ +
3	+ -
-1	answer: 1 2

Комментарий

Для корректной работы программы после каждой операции вывода данных вам необходимо делать следующие операции:

- В языке Pascal: `flush(output);`
- В C/C++: `fflush(stdout);`
- В Java: `System.out.flush();`
- В Python: `sys.stdout.flush();`

Кроме этого, не забывайте после каждой выведенной строки ставить перевод строки.

Задача J. Пещеры

Имя входного файла: `cave.in`
Имя выходного файла: `cave.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Как известно, много лет назад оборотни, предки Джейкоба и его семьи, жили в пещерах неподалеку от залива. Причем дабы избежать всевозможных проблем все они жили по одиночке.

Однако, это еще не все странности оборотней того времени. Каждый раз, на протяжении суток после очередного солнечного затмения, оборотни, возвращаясь с охоты, могли заходить в свои пещеры только по очень странному правилу, описанному далее.

Издравле сложилось, что оборотни всегда охотятся стаями, поэтому на момент их возвращения с охоты все пещеры пусты. Для удобства понимания правила, по которому оборотни заходят в пещеры, пронумеруем пещеры натуральными числами от 1 до n , начиная с самой дальней от залива пещеры.

Процесс заселения пещер состоит в следующем: сначала выбирается пещера, которую оборотни хотят заселить, или наоборот, из которой они хотят выселить оборотня. После этого, если там есть оборотень, то он выходит, а если нет, то оборотень туда заселяется.

Однако, вся сложность в том, что оборотни могут выбрать не любую пещеру, а либо пещеру с номером один, либо пещеру с номером $x + 1$, где x — первая занятая пещера.

Определите, в каком порядке оборотни могли заселять пещеры.

Формат входного файла

В единственной строке дано одно натуральное число n — количество пещер и оборотней ($1 \leq n \leq 20$).

Формат выходного файла

В первой строке выведите число k — количество совершенных действий. Во второй строке выведите последовательность действий оборотней. Если пещера освобождается, то номер пещеры должен быть выведен со знаком минус. Иначе выводите просто номер заселяемой пещеры. Количество действий не должно превышать 10^6 . Если существует несколько возможных решений задачи, то разрешается вывести любое.

Пример

<code>cave.in</code>	<code>cave.out</code>
3	5 1 2 -1 3 1