

Задача А. Шоколадка

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

64 мегабайта

Команда «Отбой» участвует в очередном марафоне по «Угадай мелодию. Rock version». Чтобы было чем подкрепиться во время игры, команда взяла с собой большую прямоугольную плитку шоколада размерами $w \times h$. У команды есть список из n пар чисел — размеры шоколадок, которые команда считает счастливыми. Прежде чем приступить к поеданию шоколадки, участники команды решили поделить имеющуюся плитку на счастливые шоколадки. Для этого они действуют следующим образом: сначала плитка шоколада ломается на 2 части по линии, строго параллельной одной из своих сторон, после чего каждую из полученных частей они могут продолжить ломать аналогичным образом.

Вам поручили определить, какое максимальное количество счастливых шоколадок команда сможет получить, действуя по данной схеме. Шоколадки, полученные поворотом счастливых, счастливыми не являются.

Формат входных данных

В первой строке входного файла заданы три целых числа w, h, n — размеры плитки шоколада и количество вариантов размера счастливых шоколадок соответственно ($1 \leq w, h \leq 300, 1 \leq n \leq w \times h$). В следующих n строках заданы пары целых чисел w_i, h_i — размеры счастливых шоколадок ($1 \leq w_i \leq w, 1 \leq h_i \leq h$).

Формат выходных данных

В единственную строку выходного файла выведите максимальное количество счастливых шоколадок, на которые можно разрезать данную плитку.

Примеры

стандартный ввод	стандартный вывод
21 11 4 10 4 6 2 7 5 15 10	15
9 12 5 1 12 2 6 3 4 4 3 6 2	9

Задача В. Упаковка символов

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Билл пытается компактно представить последовательности прописных символов от A до Z с помощью упаковки повторяющихся подпоследовательностей внутри них. Например, один из способов представить последовательность AAAAAAAAAABABABCCD - это 10(A)2(BA)B2(C) D. Он формально определяет сжатые последовательности символов и правила перевода их в несжатый вид следующим образом: Последовательность, содержащая один символ от A до Z, является упакованной. Распаковка этой последовательности даёт ту же последовательность из одного символа, повторённую $\times$ раз. Следуя этим правилам, легко распаковать любую заданную упакованную последовательность. Однако Биллу более интересен обратный переход. Он хочет упаковать заданную последовательность так, чтобы результирующая сжатая последовательность содержала наименьшее возможное число символов. Ограничения: длина исходной последовательности от 1 до 100.

Формат входных данных

В первой строке находится последовательность символов от A до Z.

Формат выходных данных

В единственной строке выводится упакованная последовательность наименьшей длины, которая распаковывается в заданную последовательность. Если таких последовательностей несколько, выведите лексикографически минимальную.

Примеры

стандартный ввод	стандартный вывод
AAAAAAAAAABABABCCD	9(A)3(AB)CCD
NEERCYESYESYESNEERCYESYESYES	2(NEERC3(YES))
A	A

Задача С. Казино

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Вновь открытое казино предложило оригинальную игру. В начале игры крупье выставляет в ряд несколько фишек разных цветов. Кроме того, он объявляет, какие последовательности фишек игрок может забирать себе в процессе игры. Далее игрок забирает себе одну из заранее объявленных последовательностей фишек, расположенных подряд. После этого крупье сдвигает оставшиеся фишки, убирая разрыв. Затем игрок снова забирает себе одну из объявленных последовательностей и так далее. Игра продолжается до тех пор, пока игрок может забирать фишки. Рассмотрим пример. Пусть на столе выставлен ряд фишек `rrrgggbbb`, и крупье объявил последовательности `rg` и `gb`. Игрок, например, может забрать фишки `rg`, лежащие на третьем и четвёртом местах слева. После этого крупье сдвинет фишки, и на столе получится ряд `rrggbbb`. Ещё дважды забрав фишки `rg`, игрок добьётся того, что на столе останутся фишки `bbb` и игра закончится, так как игроку больше нечего забрать со стола. Игрок мог бы действовать и по-другому - на втором и третьем ходах забрать не последовательности `rg`, а последовательности `gb`. Тогда на столе остались бы фишки `rrb`. Аналогично, игрок мог бы добиться того, чтобы в конце остались ряды `rrr` или `rbb`. После окончания игры полученные фишки игрок меняет на деньги. Цена фишки зависит от её цвета. Требуется написать программу, определяющую максимальную сумму, которую сможет получить игрок.

Формат входных данных

В первой строке входных данных содержится число K ($1 \leq K \leq 26$) - количество цветов фишек. Каждая из следующих K строк начинается со строчной латинской буквы, обозначающей цвет. Далее в той же строке через пробел следует целое число X_i ($1 \leq X_i \leq 150, i = 1..K$) - цена фишки соответствующего цвета. В $(K + 2)$ -ой строке описан ряд фишек, лежащих на столе в начале игры. Ряд задается L строчными латинскими буквами ($1 \leq L \leq 150$), которые обозначают цвета фишек ряда. В следующей строке содержится число N ($1 \leq N \leq 150$) - количество последовательностей, которые были объявлены крупье. В следующих N строках записаны эти последовательности. Гарантируется, что сумма длин этих N строк не превосходит 150 символов, и все они непустые.

Формат выходных данных

Выведите единственное целое число — максимальную сумму денег, которую может получить игрок.

Пример

стандартный ввод	стандартный вывод
3 v 3 l 1 u 2 luvu 3 luv vul uuu	6

Задача D. Покраска забора

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.7 секунд
Ограничение по памяти: 256 мегабайт

256 мегабайт

Мэр города Многоярославца решил построить перед своим домом забор из n деревянных досок и нанять лучшего маляра города для его покраски. Поскольку забор должен стать главной достопримечательностью города, лучший дизайнер города для каждой доски назначил тщательно выбранный цвет, в который она должна быть покрашена.

Для покраски главный маляр решил применить новейшую технологию, специально разработанную им для выполнения этого задания. Покраской забора будет заниматься специальный робот, который за один час может покрасить произвольный отрезок забора (набор соседних досок) в некоторый цвет. Поскольку задание должно быть выполнено как можно быстрее, требуется составить программу для робота, которая позволит достичь требуемой раскраски за минимальное время. Оставить какую-то из досок непокрашенной, естественно, запрещается.

Формат входных данных

В первой строке входного файла записано число n ($1 \leq n \leq 300$), где n количество досок в заборе. Вторая строка содержит строку из n символов, описывающую требуемую покраску забора. Цвета обозначаются заглавными латинскими буквами.

Формат выходных данных

В первой строке выходного файла выведите m - наименьшее возможное время покраски забора в часах. Следующие m строк должны содержать программу покраски для робота. Каждая строка должна содержать два числа l_i и r_i , а также заглавную букву латинского алфавита, задающую цвет c_i и означает, что робот должен покрасить участок забора с l_i по r_i - доску в цвет c_i (если длина забора n , должно выполняться $1 \leq l_i \leq r_i \leq n$).

Примеры

стандартный ввод	стандартный вывод
5 ABVCA	3 1 5 A 2 3 B 4 4 C
2 AA	1 1 2 A

Задача E. AliKingspress

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Для экстренного пополнения боеприпасов и вооружения агенты «Кингсман» пользуются службой «AliKingspress». Помимо запроса помощи с вооружением, можно также делать другие запросы, однако уже не бесплатно, а за бонусные баллы. Баллы можно получать каждый день, заходя в специальное приложение. В первый день пользователь получает a_1 баллов, во второй — a_2 баллов, ..., в n -й день — a_n баллов. После этого, заходя каждый день, пользователь будет все еще получать a_n бонусов. Если же пропустить один или несколько дней и не заходить в приложение, при следующем заходе начисление бонусов опять начнется с a_1 .

Эггси посчитал, что для выполнения всех дополнительных запросов, которые он хочет, нужно x бонусов. Так как он перфекционист, лишние бонусы ему не нужны, он хочет накопить их ровно x , ни больше, ни меньше. Однако сделать это нужно как можно быстрее, потому что долго ждать он не намерен. Задачу нахождения минимального количества дней, требуемого для этого, он поручил вам — своего верному программисту, пока он сам спасает мир. Помогите ему!

Формат входных данных

В первой строке содержится два числа n и x — количество различных бонусов, а также суммарное количество бонусов, которое нужно набрать Эггси ($1 \leq n \leq 100, 1 \leq x \leq 10^6$).

Во второй строке содержится n чисел a_i — размеры бонусов в зависимости от количества дней захода в приложение ($1 \leq a_i \leq 1\,000$).

Формат выходных данных

В единственной строке выведите минимальное количество дней, нужное для получения ровно x бонусов или -1 , если набрать ровно x бонусов невозможно.

Примеры

стандартный ввод	стандартный вывод
4 14 1 2 3 4	5
5 12 1 4 2 6 3	6
3 8 3 4 2	-1

Замечание

В первом примере Эггси может заходить в приложение 5 дней подряд и получить, соответственно, $1 + 2 + 3 + 4 + 4 = 14$ бонусов.

Во втором тестовом примере Эггси может зайти в приложение три дня подряд, затем пропустить один день, а затем зайти еще два дня подряд. В результате он получит $1 + 4 + 2 + 1 + 4 = 12$ бонусов и потратит на это $3 + 1 + 2 = 6$ дней.

Задача F. Подозрительная строка

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Расследуя убийство, Эркюль Пуаро нашёл подозрительную строку из маленьких латинских букв. Он предположил, что эта строка принадлежит доктору Арбэтноу, и хочет проверить свою догадку.

Доктору Арбэтноу очень нравятся палиндромы, поэтому Пуаро хочет определить, похожа ли найденная им строка на палиндром. Пуаро считает строку похожей на палиндром, если можно переставить в ней буквы так, чтобы она стала палиндромом и при этом каждая буква переместилась не более чем на одну позицию.

Помогите великому сыщику — определите, похожа ли найденная им строка на палиндром.

Формат входных данных

Во входном файле задана строка из маленьких латинских букв длины не более 10^5 .

Формат выходных данных

Выведите YES, если строка похожа на палиндром, или NO в противном случае.

Примеры

стандартный ввод	стандартный вывод
baaacba	YES
steponnopets	YES
noooo	NO

Замечание

В первом тесте из строки можно получить палиндром abacaba.

Во втором тесте строка и так является палиндромом.

Задача G. Трудности переписки

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Этим летом Джек ездил в летнюю школу в России. Там он завел много новых друзей, а также встретил красивую девушку. По возвращении домой родители подарили Джеку новый ноутбук, и теперь он всегда может быть на связи со своими новыми друзьями. Естественно, получив подарок, Джек сразу стал переписываться со своей подругой Ирой.

Отправив несколько сообщений, Джек заметил, что ноутбук, а точнее его клавиатура, работает не так, как он ожидал. В процессе ввода сообщения у ноутбука иногда внезапно срабатывает клавиша «Home», в результате чего курсор ввода перемещается в начало строки. Так, например, если у Джека в процессе ввода строки «irailikeyou» клавиша «Home» сработала после ввода букв «а» и «у», то получится строка «ouilikeyira». Джек планировал набрать строку s , нажимая по очереди на соответствующие клавиши. Закончив набор, он посмотрел на экран и увидел строку t . Теперь он хочет понять, может ли она быть результатом его ввода, если единственная неисправность его ноутбука — лишние срабатывания клавиши «Home», либо у его ноутбука есть еще проблемы. Помогите Джеку.

Формат входных данных

В первой строке задано число n — длина строк s и t ($1 \leq n \leq 5000$). Во второй строке задана последовательность маленьких латинских букв длины n — строка s . В третьей строке задана последовательность маленьких латинских букв длины n — строка t .

Формат выходных данных

Выведите «YES», если из строки s могла получиться строка t , иначе выведите «NO».

Пример

стандартный ввод	стандартный вывод
11 irailikeyou ouilikeyira	YES

Задача Н. Сумма длин путей 2

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дано дерево на n вершинах. На каждом ребре написан его вес. Требуется для каждого v посчитать сумму взвешенных длин всех путей в данном дереве, исходящих из v . Пути $\langle v, u \rangle$ и $\langle u, v \rangle$ считаются различными.

Формат входных данных

Первая строка каждого теста содержит натуральное число n — количество вершин в дереве ($1 \leq n \leq 100\,000$). Следующие $n - 1$ строк содержат по 3 натуральных числа v, u, w и описывают ребро дерева, соединяющее две вершины v и u и имеющее вес w ($1 \leq v, u \leq n, 0 \leq w \leq 10^6$).

Формат выходных данных

Выведите n чисел — требуемое в условии.

Примеры

стандартный ввод	стандартный вывод
3 1 2 1 1 3 3	4 5 7
7 1 2 1 3 4 1 7 6 1 7 5 1 7 1 1 1 4 1	9 14 17 12 15 15 10

Задача I. Максимальное подмножество вершин

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Надо бы всё-таки написать нормальную легенду, а то как-то не очень. И без легенды непонятно, почему задача так называется

Но пока легенды нет, вот формальное условие:

Вам дано дерево на n вершинах. В вершинах записаны числа a_n .

Требуется выбрать подмножество вершин с максимальной суммой a_n , чтобы никакие две соседние вершины не лежали одновременно в этом подмножестве.

Формат входных данных

В первой строке записано число n — количество вершин дерева ($1 \leq n \leq 10^6$). Во второй строке записаны через пробел n чисел v_i ($|v_i| < 10^9$), задающие значения в вершинах. В следующих $n - 1$ строках описаны ребра дерева. В $(i + 2)$ -й строке записаны номера вершин a_i, b_i ($1 \leq a_i, b_i \leq n$), означающие, что в дереве есть ребро из вершины a_i в вершину b_i .

Формат выходных данных

Выведите единственное число — максимальную искомую величину.

Пример

стандартный ввод	стандартный вывод
7 0 7 2 7 1 4 8 3 6 7 6 1 3 5 1 4 6 2 1	25

Задача J. Максимальная тройка

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Дано дерево на n вершинах. Требуется выбрать из них три так, чтобы сумма расстояний между ними была максимальна.

Формат входных данных

Первая строка каждого теста содержит натуральное число n — количество вершин в дереве ($3 \leq n \leq 1\,000\,000$). Следующие $n - 1$ строк содержат по 2 натуральных числа v, u и описывают ребро дерева, соединяющее две вершины v и u ($1 \leq v, u \leq n$).

Формат выходных данных

Выведите единственное число – максимальную сумму расстояний.

Примеры

стандартный ввод	стандартный вывод
3 1 2 1 3	4
3 1 2 2 3	4

Задача К. Логическое дерево

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Рассмотрим разновидность двоичного дерева, которую мы назовем логическим деревом. В этом дереве каждый уровень полностью заполнен, за исключением, возможно, последнего (самого глубокого) уровня. При этом все вершины последнего уровня находятся максимально слева. Дополнительно, каждая вершина дерева имеет ноль или двоих детей.

Каждая вершина дерева имеет связанное с ней логическое значение (1 или 0). Кроме этого, каждая внутренняя вершина имеет связанную с ней логическую операцию («И» или «ИЛИ»). Значение вершины с операцией «И» — это логическое «И» значений её детей. Аналогично, значение вершины с операцией «ИЛИ» — это логическое «ИЛИ» значений её детей. Значения всех листьев задаются во входном файле, поэтому значения всех вершин дерева могут быть найдены.

Наибольший интерес для нас представляет корень дерева. Мы хотим, чтобы он имел заданное логическое значение v , которое может отличаться от текущего. К счастью, мы можем изменять логические операции некоторых внутренних вершин (заменить «И» на «ИЛИ» и наоборот).

Дано описание логического дерева и набор вершин, операции в которых могут быть изменены. Найдите наименьшее количество вершин, которые следует изменить, чтобы корень дерева принял заданное значение v . Если это невозможно, то выведите строку «IMPOSSIBLE» (без кавычек).

Формат входных данных

В первой строке входного файла находятся два числа n и v ($1 \leq n \leq 10\,000$, $0 \leq v \leq 1$) — количество вершин в дереве и требуемое значение в корне соответственно. Поскольку все вершины имеют ноль или двоих детей, то n нечётно. Следующие n строк описывают вершины дерева. Вершины нумеруются от 1 до n .

Первые $(n-1)/2$ строк описывают внутренние вершины. Каждая из них содержит два числа — g и c , которые принимают значение либо 0, либо 1. Если $g = 1$, то вершина представляет логическую операцию «И», иначе она представляет логическую операцию «ИЛИ». Если $c = 1$, то операция в вершине может быть изменена, иначе нет. Внутренняя вершина с номером i имеет детей $2i$ и $2i+1$.

Следующие $(n+1)/2$ строк описывают листья. Каждая строка содержит одно число 0 или 1 — значение листа.

Формат выходных данных

В выходной файл выведите ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
9 1 1 0 1 1 1 1 0 0 1 0 1 0 1	1
5 0 1 1 0 0 1 1 0	IMPOSSIBLE

Задача L. НВП

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.25 секунд
Ограничение по памяти: 256 мегабайт

Числовая последовательность задана рекуррентной формулой: $a_{i+1} = (k \cdot a_i + b) \bmod m$. Найдите её наибольшую возрастающую подпоследовательность. Если таких последовательностей несколько, можно вывести любую.

Формат входных данных

Программа получает на вход пять целых чисел: длину последовательности n ($1 \leq n \leq 10^5$), начальный элемент последовательности a_1 , параметры k, b, m для вычисления последующих членов последовательности ($1 \leq m \leq 10^4, 0 \leq k < m, 0 \leq b < m, 0 \leq a_1 < m$).

Формат выходных данных

На первой строке выходного файла вы должны вывести количество чисел в найденной вами наибольшей возрастающей подпоследовательности. На следующей строке выведите элементы подпоследовательности, разделяя их пробелами.

Примеры

стандартный ввод	стандартный вывод
5 41 2 1 100	3 41 67 71
7 1 2 1 10	4 1 3 5 7
7 2 2 1 10	3 1 3 5

Задача М. Линар и банк

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Линар хочет положить для сохранности в банк «Центр-инвест» n мешков с деньгами, так как там самые выгодные условия. У Линара есть бесконечное количество бронемашин, в каждую машину вмещается k мешков. В i -м мешке c_i единиц денег.

Мешки расположены подряд, когда Линар будет складывать их, он положит первые k мешков в первую бронемашину, следующие k мешков во вторую бронемашину и т.д.

Прежде чем складывать мешки Линар хочет поменять их порядок так, чтобы сумма денег в первой бронемашине была максимальна, сумма денег во второй бронемашине была максимальна среди оставшихся мешков и т.д.

Линар может выбрать мешок и поставить в любое место в последовательности.

Скажите, какое минимальное кол-во действий надо сделать Линару, чтобы увезти все деньги.

Формат входных данных

Первая строка содержит два целых числа n и k ($1 \leq n, k \leq 10^5$) — количество мешков и вместительность бронемашин.

Во второй строке по n целых чисел c_i ($1 \leq c_i \leq 10^5$) — количество денег в каждом мешке.

Формат выходных данных

Скажите какое минимальное кол-во действий надо сделать Линару, чтобы увезти все деньги.

Примеры

стандартный ввод	стандартный вывод
6 2 10 4 5 3 11 100	3
5 2 100 78 72 95 2	1
6 2 72 6 52 52 99 53	3
7 2 78 29 5 93 31 10 29	2

Задача N. Футбол

Имя входного файла: `football.in`
Имя выходного файла: `football.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Футбольный клуб «Инжир» не имеет проблем с финансированием, поэтому приобрел N полевых игроков. После долгих месяцев тренировок и тестов тренерский коллектив подробно изучил каждого игрока и собирается выбрать оптимальный стартовый состав.

Каждый игрок характеризуется тремя целыми неотрицательными числами: d_i — умение играть в защите, m_i — умение играть в полузащите и a_i — умение играть в нападении. Играть было решено по схеме 5-3-2. Это означает, что в основном составе будет 5 игроков защиты, 3 полузащитника и 2 нападающих. Цель тренерского штаба — выбрать 10 игроков так, чтобы сумма соответствующих умений была максимальна (для игрока обороны значение имеет только его умение играть в защите, аналогично для других позиций). Разумеется, каждый игрок может играть не более чем на одной позиции. Ваша задача — помочь тренерскому штабу подобрать оптимальный состав.

Формат входных данных

В первой строке входного файла находится одно целое число N — количество игроков в клубе. Следующие N строк файла содержат по три целых числа каждая d_i , m_i и a_i — умения футболиста с номером i играть в обороне, полузащите и нападении соответственно.

Формат выходных данных

Выходной файл должен содержать ровно три строки. Первая строка должна содержать пять чисел — номера игроков обороны. Вторая строка должна содержать ровно три числа — номера игроков полузащиты. Третья строка должна содержать ровно два числа — номера игроков нападения. Если оптимальных ответов несколько, можно вывести любой.

Примеры

football.in	football.out
10 1 2 3 2 3 4 3 4 5 4 5 6 5 6 7 6 7 8 7 8 9 8 9 10 9 10 1 10 1 2	10 9 8 7 6 3 4 5 1 2
10 1 5 1 3 2 2 3 3 5 2 1 3 4 4 5 2 0 2 1 1 1 4 5 1 1 3 3 4 5 4	10 5 2 6 7 1 8 9 3 4

Замечание

Задача О. Польшар и Подарки

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Рождество! Польшар и его друзья будут дарить друг другу подарки. Всего шаров n . Каждый шар должен подарить подарок ровно одному другому шару в соответствии с некоторой перестановкой p , $p_i \neq i$ для всех i .

К сожалению, шары забывчивы. Мы знаем, что ровно k шаров забудут принести свои подарки. Шар номер i получит подарок, если будут выполнены следующие два условия:

1. Шар номер i должен принести свой подарок.
2. Шар x такой, что $p_x = i$, должен принести свой подарок.

Какое минимально и максимально возможное число шаров, которые **не** получают свой подарок, если ровно k шаров забудут принести свой подарок?

Формат входных данных

В первой строке находится два целых числа n и k ($2 \leq n \leq 10^6$, $0 \leq k \leq n$) — общее число шаров и число шаров, которые забудут подарки.

Во второй строке находится перестановка p целых чисел от 1 до n , где p_i — номер шара, которому должен дать подарок шар номер i . Для всех i выполняется $p_i \neq i$.

Формат выходных данных

Выведите два числа — минимально и максимально возможное число шаров, которые **не** получают подарков, соответственно.

Примеры

стандартный ввод	стандартный вывод
5 2 3 4 1 5 2	2 4
10 1 2 3 4 5 6 7 8 9 10 1	2 2

Замечание

В первом примере, если первый и третий шары забудут принести подарок, то они же и будут единственными, кто не получит подарка. Поэтому минимальный ответ равен 2. Однако, если первый и второй шары забудут, то только пятый шар получит подарок. Поэтому максимальный ответ равен 4.