

Задача А. The Elder Trolls V: Путь Истины

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вася прошел игру «The Elder Trolls V» несколько раз и решил перейти к созданию модов — заменить модели персонажей, оружия и прочего на собственные разработки. Для этого он правит файлы игры, заменяя пути к ресурсам. Когда Вася попытался показать свой мод другу — его ждало разочарование: игра отказалась загружать новые модели. Выяснилось, что друг установил игру в другой каталог, а Вася указал в файлах абсолютные пути к ресурсам. К счастью, игра понимает и пути, указанные относительно текущего каталога, но Васе не хочется править все файлы вручную. Вам нужно написать программу, которая поможет ему в этом.

Вася и его друзья используют операционную систему Unix. В ней файлы вложены в каталоги, каталоги, в свою очередь, могут быть вложены в другие каталоги. Вершиной иерархии является корневой каталог. Абсолютным путем к файлу/каталогу называется цепочка имен каталогов, в которые он вложен, начинающаяся с символа '/', имена каталогов в цепочке разделяются этим же символом; цепочка заканчивается именем этого файла/каталога. Например, если каталог **b** вложен в каталог **a**, который вложен в корневой каталог, то абсолютный путь к каталогу **b** записывается как `/a/b`. Абсолютный путь к корневому каталогу состоит только из символа '/'. Название каталога может содержать маленькие латинские буквы, цифры и дефис.

Относительный путь задается относительно известного каталога. Например, путь к файлу `/b/c/a` относительно каталога `/b`, записывается как `c/a`. Переход в родительский каталог обозначается специальным именем `..` (две точки). Например, путь к каталогу `/a/b/c` относительно каталога `/a/b/d` записывается как `../c`.

Формат входного файла

На вход программе подаются две строки. Каждая строка представляет собой корректный абсолютный путь к каталогу и состоит из символов 'a'-'z', '0'-'9', '-' и '/'. Пути **не** содержат пробелов и переходов к родительским каталогам `..`. Длина каждой строки не превышает 1000 символов. Гарантируется, что два данных абсолютных пути различны.

Формат выходного файла

Выведите одну строку: кратчайший путь ко второму каталогу относительно первого. Кратчайший путь содержит наименьшее возможное количество названий каталогов и переходов в родительский каталог.

Примеры

stdin	stdout
<code>/home/vasya/tes-v/data/sounds/music</code> <code>/home/vasya/tes-v/data/images/portraits</code>	<code>../../images/portraits</code>
<code>/home/user</code> <code>/home/user/photo</code>	<code>photo</code>

Задача В. Rangy Dribs: Ожидание

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вася любит играть в «Rangy Dribs» на своем телефоне когда чего-то ждет. Однажды он сидел в зале ожидания в аэропорту, играя как обычно, но у него сел аккумулятор. От нечего делать он стал смотреть по сторонам и заметил любопытный факт: несмотря на то, что в зале оставалось довольно много свободных мест, люди не спешили занимать их, а уходили в другие залы. Вася понял, что люди не любят, когда в соседних креслах сидят незнакомые люди, и предпочтут скорее найти другое место, чем сесть в кресло, если хотя бы одно из соседних кресел уже занято незнакомцем.

Зал ожидания представляет собой длинный ряд из N кресел, пронумерованных от 1 до N . Некоторые кресла уже заняты. Васе стало любопытно, сколькими способами K новоприбывших людей смогут разместиться в зале ожидания, если они не знакомы как между собой, так и с уже сидящими людьми.

Формат входного файла

В первой строке даны три числа, разделенные пробелами: N ($1 \leq N \leq 10\,000$) — количество кресел в зале, M ($0 \leq M \leq N$) — количество уже занятых кресел, K ($0 \leq K \leq 1\,000$) — количество новоприбывших. Если $M > 0$, то во второй строке записаны M чисел a_i ($1 \leq a_i \leq N$), разделенных пробелами — номера занятых кресел. Все номера различны.

Формат выходного файла

Выведите одно число — сколькими способами K человек могут разместиться в этих N креслах, не занимая места рядом с незнакомцами. Поскольку это число может быть большим, выведите его остаток от деления на 1 000 007.

Примеры

stdin	stdout
5 1 1 2	2
5 2 1 3 2	1
4 0 2	6

Примечания

В первом тесте возможны следующие раскладки: (-*-1-); (-*--1).

Во втором возможна только следующая: (-**-1).

В третьем тесте возможны следующие: (1-2-); (1--2); (-1-2); (2-1-); (2--1); (-2-1).

Здесь свободные кресла обозначены как -, изначально занятые — как *, а кресла, занятые новоприбывшими — их номерами (1,2).

Задача С. SplineEdge: Инвентарь

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вася играет в популярную RPG SplineEdge и ему вечно не хватает места в инвентаре персонажа.

Инвентарь имеет ограниченное количество слотов для размещения вещей, и Васе приходится идти на хитрость: в часть слотов он кладет сумки, которые могут вместить в себя дополнительные предметы.

Проблемы начинаются, когда Васе нужно что-то найти в своём инвентаре. По причине безнадёжной рассеянности Вася понятия не имеет, куда положил драгоценный артефакт S . Чтобы морально подготовиться к многочасовому процессу поиска, Вася хочет узнать, какое минимальное и максимальное время может занять отыскание артефакта.

Опишем виртуальную трагедию формально. Инвентарь Васи содержит N предметов, каждый из которых может вмещать некоторые другие предметы. Предмет i характеризуется своим номером id_i и номером предмета внутри которого он непосредственно находится idp_i ; если предмет лежит в основном инвентаре, то $idp_i = 0$.

На экране Васи может быть открыто только одно окно — с содержимым текущей сумки. В момент времени 0 открыт основной инвентарь. Вася пользуется следующим алгоритмом поиска:

1. Если открыто содержимое сумки, в которой лежит предмет S , Вася тут же узнает его, поиск считается оконченным.
2. Если открытая сумка не содержит других непустых сумок, в которых ещё не смотрел Вася, он закрывает текущее окно и попадает в сумку-родитель.
3. Если в текущем окне ещё есть непустые сумки, Вася наугад выбирает одну из тех, где он ещё не смотрел, открывает её содержимое и повторяет описанный процесс для неё.

Так как игра сильно тормозит, процессы открытия и закрытия окон с содержимым сумок занимают ровно 1 секунду.

Требуется найти время, необходимое для поиска предмета с $id = S$ в лучшем и худшем случае.

Формат входного файла

Первая строка содержит целое число N ($1 \leq N \leq 100000$) — количество предметов в инвентаре.

Следующие N строк содержат описание предметов. Строка $i + 1$ содержит id_i (номер предмета) и idp_i (номер предмета, в котором он находится), разделенные пробелом.

Последняя строка ввода содержит число S — номер предмета, который ищет Вася.

Номера предметов уникальны и лежат в пределах от 1 до 10^6 включительно.

Гарантируется, что среди предметов есть предмет с номером S и никакая пара предметов не указана дважды. Все предметы лежат либо в основном инвентаре, либо в одной из сумок внутри инвентаря.

Формат выходного файла

В единственной строке выведите через пробел два числа T_1 и T_2 , где T_1 — время поиска в лучшем случае, а T_2 — в худшем.

Примеры

stdin	stdout
5 1 0 2 0 3 1 4 2 5 2 5	1 3
1 1 0 1	0 0
9 1 0 2 1 3 0 40 3 50 3 100 40 200 50 1000 100 2000 200 2	1 11

Примечания

Примечание 1:

В первом примере, очевидно, в лучшем случае Вася откроет вторую сумку и сразу же найдет искомый предмет 5. На открытие сумки уходит 1 секунда, так что $T_1 = 1$. Если же Вася сначала откроет первую сумку, ему придется потратить ещё 2 секунды вхолостую — на открытие и закрытие соответствующего окна, $T_2 = 1 + 2 = 3$.

Поясним третий пример. В лучшем случае Васе нужна одна секунда на то, чтобы открыть сумку с искомым предметом — первую, $T_1 = 1$. Если же Вася решит поступить иначе, ему придется исследовать все непустые, лежащие внутри третьей сумки (включая её саму). Таких сумок 5 штук, значит, дополнительно необходимо 10 секунд на открытие и закрытие каждого окна. Плюс одна секунда на то, чтобы исследовать первую сумку. Итого, $T_2 = 11$.

Примечание 2:

Входные данные могут иметь размер больше 1МБ, не рекомендуется использовать объект *cin* для чтения входных данных. Чтение большого объема данных с использованием *cin* требует достаточно много процессорного времени, что может привести к вердикту «*Time Limit Exceeded*».

Задача D. X-Dom: Железная дорога

Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Вася играет в X-Dom: The Distance Unknown. В этой игре Васе предстоит спасти Землю от инопланетного вторжения. Для этого он объединил в союз все страны от Германии до Китая. Для быстрого перемещения войск между ними была построена прямая железная дорога. Однако, пришельцы решили разрушить транспортную систему Васи и отправили на Землю N энергетических станций. Далее был запущен механизм уничтожения железной дороги. Происходило это следующим образом: для каждой пары станций в середине отрезка, соединяющего их, создавался мощный заряд. Далее из каждого такого заряда выпускалось два разрушающих луча, лучи летели по прямой, ортогональной соответствующему двум станциям отрезку, и были направлены в противоположные стороны. Если луч пересекал железную дорогу, то он повреждал ее в точке пересечения. Вам необходимо оценить масштабы разрушений, а именно найти расстояние между двумя наиболее удаленными поврежденными участками дороги.

Формат входного файла

В первой строке дано целое число N ($2 \leq N \leq 75000$) — количество инопланетных станций. В каждой из следующих N строк даны два целых числа X_i и Y_i , по модулю не превосходящие 100000, — абсцисса и ордината положения соответствующей станции. Все станции находятся в различных точках и не лежат на оси OX . Железная дорога Васи идет вдоль прямой $Y = 0$ и является бесконечной в обоих направлениях. Гарантируется, что хотя бы один из полученных разрушающих лучей пересекает ось OX . Гарантируется, что ни один из разрушающих лучей не совпадает с осью OX .

Формат выходного файла

Вывести единственное вещественное число с абсолютной или относительной погрешностью не более 10^{-6} — искомое расстояние между двумя наиболее удаленными поврежденными участками дороги.

Примеры

stdin	stdout
3 0 1 -2 5 2 5	14.000000

Примечания

Примечание 1:

В примере заряды создаются в точках $(-1, 3)$, $(0, 5)$, $(1, 3)$, а лучи повреждают дорогу в точках $(-7, 0)$, $(0, 0)$ и $(7, 0)$ соответственно.

Примечание 2:

Входные данные могут иметь размер больше 1МБ, не рекомендуется использовать объект *cin* для чтения входных данных. Чтение большого объема данных с использованием *cin* требует достаточно много процессорного времени, что может привести к вердикту «*Time Limit Exceeded*».

Задача E. Idablo 3: Читерство

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вася играет в Idablo 3 — очень популярную игру жанра hack and slash. После нескольких дней непрерывных боев он смог заработать своим персонажем кругленькую сумму виртуальных золотых! Но, все же, Васе не хватает денег, чтобы купить Меч Ангела — очень редкое и смертоносное двуручное оружие. Поэтому он решил взломать игру и заработать на покупку желанного предмета.

Разработчики Idablo 3 встроили в игру специальную систему защиты от читерства. Как известно, сохранения Idablo 3 хранятся на серверах компании-разработчика, а не на компьютере пользователя. Однако, Вася нашел специальный чит-код, который позволяет, нажав определенную комбинацию клавиш во время игры, дать персонажу любую сумму денег.

Казалось бы — мечта сбылась и Меч Ангела уже в руках! Но нет! Хитрые разработчики встроили в игру еще одну систему защиты, которая заключается в следующем. При внезапном изменении количества денег у персонажа (как в случае с применением чит-кода) выполняется проверка, в случае невыполнения которой игрок признается читером и его персонаж удаляется навсегда...

Проверка состоит в следующем. Пусть исходное количество золотых у персонажа было равно N , а после внезапного изменения стало равно M . Тогда игрок будет признан читером, если выполняется хотя бы одно из двух условий:

- M не может быть получено из N перестановкой цифр и, возможно, удалением некоторых из них. Например, число 123 может быть получено из 213, а 223 не может быть получено из 352;
- в записи числа M есть две одинаковые подряд идущие цифры.

Естественно, Вася хочет получить как можно больше виртуальных денег, при этом, чтобы его персонажа не удалили. Помогите ему узнать, какое максимальное количество золотых может быть у его персонажа после применения чит-кода.

Формат входного файла

Единственная строка входных данных содержит целое число N ($1 \leq N \leq 10^{1000}$) — исходное количество золотых у персонажа Васи.

Формат выходного файла

Выведите единственное число — максимальное количество золотых M , которое может иметь персонаж Васи после применения чит-кода.

Примеры

stdin	stdout
123	321
999	9

Примечания

В первом примере можно составить 6 различных чисел: 123, 132, 213, 231, 312, 321. Все они являются допустимыми, так как не содержат двух одинаковых подряд идущих цифр. Максимальное из них это 321.

Во втором примере можно составить 3 различных числа: 9, 99, 999. Но только 9 подходит, так как в других содержатся две подряд идущие девятки.

Задача F. PaintMan: Малярия

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вася играет в популярную логическую игру PaintMan.

Игровое поле представляет собой прямоугольник $N \times M$, состоящий из N строк и M столбцов. Персонаж Васи может ходить в любую клетку, имеющую общую сторону с текущей (таким образом, за ход он может переместиться на одну клетку вверх, вниз, влево или вправо). При этом, он не может выходить за пределы игрового поля. Каждая клетка покрашена либо в черный, либо в белый цвет. Как только персонаж Васи ступает на клетку, то он перекрашивает ее в противоположный цвет (с белого на черный и с черного на белый соответственно).

Необходимо вывести действия, которые необходимо сделать персонажу Васи, чтобы перекрасить все клетки игрового поля в один цвет. Количество шагов не обязательно должно быть минимальным, но не должно превышать 100000.

Клетки в столбцах и рядах нумеруются с 1. Левая верхняя клетка имеет координату $(1, 1)$, правая нижняя — (N, M) . В начальный момент времени персонаж Васи находится в клетке (i, j) .

Формат входного файла

Первая строка входных данных содержит два целых числа N и M , разделенные пробелом ($1 \leq N, M \leq 50$) — размеры игрового поля.

Вторая строка входных данных содержит два целых числа i и j , разделенные пробелом ($1 \leq i \leq N, 1 \leq j \leq M$) — начальные координаты персонажа Васи.

Последующие N строк состоят из M символов каждая: «#» обозначает клетку, покрашенную в черный цвет, а «.» — клетку белого цвета.

Формат выходного файла

Единственная строка — список команд персонажу Васи, состоящая из букв «U», «D», «L», «R»:

- «U» соответствует ходу на 1 клетку вверх;
- «D» соответствует ходу на 1 клетку вниз;
- «L» соответствует ходу на 1 клетку влево;
- «R» соответствует ходу на 1 клетку вправо.

Если решений несколько, то выведите любое из них.

Примеры

stdin	stdout
2 3 2 1 .# .##	RRU
2 4 1 1 ##.. ##.#	DURDRRL

Примечания

Во втором примере игровое поле будет выглядеть следующим образом после очередного хода:

##.. ⇒ ##.. ⇒ .#.. ⇒ ⇒ ⇒ ⇒ ⇒
##.# ⇒ .#.# ⇒ .#.# ⇒ .#.# ⇒ ...# ⇒ ...# ⇒ ...# ⇒ ...#

Задача G. MindCraft: Вертолетная площадка

Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Вася играет в MindCraft. Вася выбрал себе поле размером $N \times M$ клеток, на каждой клетке которого может стоять несколько единичных кубиков. Количество кубиков, стоящих на определенной клетке, назовем высотой этой клетки. За одну секунду Вася может поставить на любую клетку один кубик или убрать с клетки один кубик (если на ней было положительное число кубиков), т.е. изменить высоту клетки на 1. Вася хочет, чтобы на его поле появилась вертолетная площадка размером $H \times W$ клеток. Для этого Васе нужно, чтобы на его поле был прямоугольник размером $H \times W$, все клетки которого имели бы одинаковую высоту. В этом случае площадка получится ровной и Вася будет крайне доволен. Ваша задача — определить, за какое наименьшее время Вася сможет построить такую площадку.

Формат входного файла

В первой строке находятся два целых числа N и M ($1 \leq N, M \leq 500$) — размеры поля, на котором играет Вася. В следующей строке находятся два целых числа H и W ($1 \leq H \leq 20, 1 \leq W \leq 200, H \leq N, W \leq M$) — размеры вертолетной площадки. Далее следуют N строк по M чисел в каждой — начальные высоты клеток. Высоты клеток неотрицательные и не превосходят 100000.

Формат выходного файла

Вывести единственное число — минимальное время, за которое Вася сможет подготовить вертолетную площадку.

Примеры

stdin	stdout
2 3 1 2 1 2 3 4 5 6	1
2 3 1 2 1 6 3 4 2 5	2

Примечания

Примечание 1:

В первом примере площадку можно строить в любом месте — всегда нужно будет добавить (или убрать) один кубик.

Во втором примере Вася может убрать из первой клетки второго ряда два кубика, получив тем самым площадку размером 1×2 высоты 2.

Примечание 2:

Входные данные могут иметь размер больше 1МБ, не рекомендуется использовать объект *cin* для чтения входных данных. Чтение большого объема данных с использованием *cin* требует достаточно много процессорного времени, что может привести к вердикту «*Time Limit Exceeded*».

Задача Н. World Of WorkCraft: Раздел добычи

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вася с друзьями играет в World Of WorkCraft. Как известно, это одна из самых популярных онлайн-ролевых игр (MMORPG), в которой игроки могут объединяться и выполнять задания совместно!

После очередного успешного рейда группа игроков, в которую входит и Вася, решила разделить полученную добычу. За выполнение задания ребята могут выбрать ровно одну из возможных N наград, каждая из которых представляет собой совокупность объектов одного типа: набор одинаковых вещей, деньги, опыт и т.д. Каждая награда характеризуется количеством элементов в наборе. Например, наградами могут быть 1000 единиц опыта или 2000 стрел, или 500 золотых и т.д.

Так как некоторые игроки показали себя не с лучшей стороны (погибли в первые секунды боя, например), то они не примут участие в дележе. Однако, ребята очень честные, поэтому они хотят поделить добычу таким образом, чтобы все, кто примут участие в дележе, получили поровну, и, при этом, вся добыча будет распределена. Так как друзья еще не определили, кто получит часть награды, а кто нет, то они хотят выбрать ее таким образом, чтобы количество вариантов разделить добычу на равные доли было наибольшим.

Если несколько различных наград удовлетворяют условию, то необходимо выбрать ту из них, в которой количество элементов в наборе максимально. Гарантируется, что количество элементов во всех наборах различно.

Формат входного файла

Первая строка содержит целое число N ($1 \leq N \leq 1000$) — количество различных наград.

Вторая строка содержит N целых чисел A_i ($1 \leq A_i \leq 10^9$) — количество элементов в награде i . Все числа разделены пробелами. Награды нумеруются с 1.

Формат выходного файла

Единственное целое число — номер набора, который необходимо выбрать.

Примеры

stdin	stdout
3 7 12 15	2
2 5 11	2

Примечания

В первом примере первую награду можно разделить на 1 и 7 частей, вторую — на 1, 2, 3, 4, 6 и 12 частей, а третью — на 1, 3, 5 и 15 частей. Больше всего различных вариантов для второй награды.

Во втором примере первую награду можно разделить двумя способами: на 1 и 5 частей, а вторую — на 1 и 11 частей соответственно. Но так как 11 больше чем 5, то надо выбрать вторую награду.

Задача I. Last Effect 3: Операция 'Сохранение'

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вася играет в Last Effect 3. Подробности геймплея этой игры в данной задаче неважны, отметим лишь, что Вася достиг значительных успехов в прохождении.

Теперь Вася хочет сохранить текущую игру, чтобы не потерять нажитые непосильным трудом достижения. В меню сохранения было предложено автоматически сгенерированное имя игры — непустая строка S . Однако, Вася стремится к прекрасному и хочет, чтобы имя сохраненной игры было палиндромом (то есть читалось слева-направо и справа-налево одинаково). Он скопировал строку S в редактор Microsoft Word и преступил к изменению строки. Пользуясь инструментом *замена*, Вася за одну операцию может поменять в текущей строке все вхождения некоторой буквы x на другую букву y . Вам нужно определить, за какое наименьшее число операций Вася сможет получить из строки S палиндром.

Формат входного файла

На вход программе подается строка S , состоящая из строчных букв латинского алфавита. Длина строки не менее 1 и не более 1000 символов.

Формат выходного файла

Вывести единственное число — наименьшее количество операций, требуемых Васе.

Примеры

stdin	stdout
aba	0
aaabb	1
abcd	2

Примечания

В первом примере строка уже является палиндромом, поэтому Васе можно ничего не делать, ответ 0.

Во втором примере необходимо заменить буквы 'a' на 'b' или наоборот.

В третьем примере необходимо заменить 'a' на 'd' и 'b' на 'c'.

Задача J. Unknown Game: Скобки

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вася играет в игру. А мы займемся более серьезными делами.

Вы, наверное, знаете, что такое правильная скобочная последовательность (ПСП). А если не знаете, то это арифметическое выражение со скобками, из которого стерли все, кроме скобок — так что каждой открывающей скобке соответствует закрывающая. Более формально

- '()' — правильная скобочная последовательность;
- если a — ПСП, то ' (a) ' — тоже ПСП;
- если a и b — ПСП, то ' ab ' (цепочка) — тоже ПСП.

Например '()', '(())', '()((()))' — ПСП, а '(', '()' и '())()' — нет.

Будем упорядочивать скобки по следующим правилам:

1. каждая скобка располагается на отдельной строке;
2. первая открывающая скобка выводится без отступа, первым символом строки;
3. если открывающая скобка вложена в другую ('(...))...'), то вторая должна быть выведена с отступом на два пробела больше, чем первая;
4. закрывающая скобка выводится с тем же отступом, что и соответствующая ей открывающая.

Формат входного файла

На вход подается единственная строка, содержащая только символы '(' и ')'. Гарантируется, что строка содержит хотя бы два и не более 1000 символов и является правильной скобочной последовательностью.

Формат выходного файла

Выведите отформатированную в соответствии с правилами последовательность.

Примеры

stdin	stdout
((()())()	(..(..) ..(..)) ()
()	()

Примечания

Для наглядности в примерах вывода программы пробелы заменены на точки '.'. Ваша программа должна выводить пробелы.

Задача К. Logo: Кубики

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вася играет в кубики Logo. Или, если быть точным, играл до недавнего времени. Теперь он перешел на другие игрушки — сборку домашнего сервера для хранения видеороликов сражений мастеров WordCroft. У Васи есть N жестких дисков и M блоков питания для них. Чтение интернета навело его на мысль собрать корпус сервера из кубиков Logo — у него осталось K штук со времен начальной школы. Вася знает, что для сборки корзины для диска ему нужно A кубиков, а для сборки крепления для блока питания — B кубиков. Сервер будет состоять только из корзин и креплений. Сколько денег понадобится Васе, чтобы докупить недостающие детали, если компания Logo продает комплект из 650 кубиков за 30 долларов?

Формат входного файла

В первой строке через пробел записаны два числа: N — количество жестких дисков, M — количество блоков питания ($0 \leq N, M \leq 100$). Во второй строке, также через пробел, записаны три числа: K — количество уже имеющихся у Васи кубиков, A — количество кубиков, нужное для сборки корзины жесткого диска, B — количество кубиков, нужное для сборки крепления для блока питания ($0 \leq K, A, B \leq 1000$).

Формат выходного файла

Выведите единственное число: сумму в долларах, которую Вася должен потратить, чтобы докупить недостающие кубики.

Примеры

stdin	stdout
5 5 25 3 4	30

Примечания

В примере Васе нужно 35 кубиков для того, чтобы собрать сервер. Однако, у него есть только 25 кубиков, поэтому ему придется купить еще один комплект за 30 долларов.