



А. Неполный API

Ограничение на время работы: для C и C++ — 1 секунда, для Java и Kotlin — 3 секунды
Ограничение на использование памяти: 256МБ

Боря Ифзэнэлсов учится на втором курсе в X-OSPU (eXtreme Open Source Programming University). Недавно у его группы была лекция по компьютерной графике. Преподаватель рассказывал про API функции, которые позволяют рисовать на экране точки, отрезки, прямоугольники, выполнять заливку и так далее.

В качестве домашнего задания студенты получили лабораторную: дана картинка, и нужно написать программу, которая воспроизведёт её на экране компьютера. Программа рисует в области экрана, размер которой точно совпадает с размером картинки, и обязана перекрасить все пиксели в этой области. Картинка дана с точным указанием цвета каждого пикселя, и воспроизвести её нужно также с точностью до цвета каждого пикселя.

Боря был уверен, что легко справится с заданием, но поздно ночью в день сдачи лабораторной обнаружилась проблема. Оказалось, что конспект лекции пришел в негодность во время практики по истории, когда Боря вместе с другими ребятами с задних парт занимался воспроизведением исторических сражений при помощи разобранных ручек и смоченных кусочков бумаги.

После сражений уцелели только две функции: `DrawHorLine(y, color)` и `DrawVerLine(x, color)`. Обе функции рисуют линию заданного цвета толщиной 1 пиксель через всю область, первая — горизонтальную, а вторая — вертикальную.

Поскольку было уже слишком поздно, чтобы просить конспект у одногруппников, которые изображали нейтральные государства, и поскольку искать документацию в Google учат только на третьем курсе, Боря решил рисовать теми функциями, которые у него есть. Помогите ему составить программу, которая выведет на экран требуемую картинку, используя лишь две указанные функции (если это, конечно, возможно). Если таких программ есть несколько, Боре подойдет любая.

Входные данные

В первой строке находятся два числа: W и H ($1 \leq W, H \leq 200$) — ширина и высота картинки, которую необходимо нарисовать. Далее следует H строк, в каждой из которых находится по W целых чисел C_{ij} ($1 \leq C_{ij} \leq 10^9$), разделённых пробелами, — цвета пикселей картинки.

Выходные данные

Если нарисовать картинку указанными функциями невозможно, выведите только число «-1». Иначе, в первой строке выведите число Q ($1 \leq Q \leq 1000$) — количество команд в программе, а в последующих Q строках — сами команды в порядке выполнения. Формат команд следующий:

- `hor Y C` — закрасить всю строку пикселей Y цветом C ($1 \leq Y \leq H$, $1 \leq C \leq 10^9$)
- `ver X C` — закрасить весь столбец пикселей X цветом C ($1 \leq X \leq W$, $1 \leq C \leq 10^9$)

Пример

Входные данные	Результат
3 3 1 2 1 2 2 2 1 2 1	4 hor 1 1 hor 2 2 hor 3 1 ver 2 2
2 2 1 2 2 1	-1



В. Носки

Ограничения: время работы — 1 секунда, использование памяти — 256МБ

В мешке находится n пар носков. Какое минимальное количество носков нужно вынуть, не заглядывая в мешок, чтобы среди выбранных оказалась хотя бы одна пара?

Входные данные

Одно число n ($1 \leq n \leq 2\,147\,483\,647$) — количество пар носков.

Выходные данные

Выведите одно число — минимальное количество носков, которые нужно вынуть.

Пример

Входные данные	Результат
1	2



С. Нечетный набор

Ограничения: время работы — 1 секунда, использование памяти — 256МБ

Представьте целое неотрицательное число N в виде суммы минимального количества нечетных чисел.

Входные данные

В единственной строке задано целое неотрицательное число N ($0 \leq N \leq 10^{18}$).

Выходные данные

В первой строке выведите количество нечетных чисел M . Во второй строке выведите M нечетных чисел, вмещающихся в знаковое 64-битное целое (`int64_t` в C++).

Если ответов несколько, выведите любой.

Пример

Входные данные	Результат
0	0
8	2 3 5



D. Побудова рядка

Обмеження на час роботи: для C та C++ — 1 секунда, для Java та Kotlin — 2 секунди
Обмеження на використання пам'яті: 256МБ

Задано ціле число n . Виведіть рядок із малих літер латинського алфавіту такий, що його довжина не перевищує 10^5 , а кількість його підрядків, які є паліндромами, дорівнює n . Паліндром — це рядок, який однаково читається як справа наліво, так і зліва направо.

Вхідні данні

В єдиному рядку міститься ціле число n ($1 \leq n \leq 10^9$).

Вихідні данні

В єдиному рядку виведіть шуканий рядок $s_1 s_2 \dots s_m$ ($1 \leq m \leq 10^5$). Якщо існує декілька відповідей, виведіть будь-яку із них.

Приклад

Вхідні данні	Результат
4	aba
1	c

В першому прикладі підрядками-паліндромами є підрядки $[1;1]$ («a»), $[2;2]$ («b»), $[3;3]$ («a») та $[1;3]$ («aba»).



Е. Эмгыр и взлом сети

Ограничение на время работы: для С и С++ — 1 секунда, для Java и Kotlin — 12 секунд
Ограничение на использование памяти: 256МБ

Информатика — не самый любимый предмет Эмгыра, поэтому, на одном из уроков информатики Эмгыр решил внести разнообразие — взломать компьютер преподавателя.

Компьютеры связаны между собой необычным способом, в одностороннем порядке, то есть, если от компьютера А есть связь с компьютером Б, то от Б нет связи с А. Также известно, что в сети нет циклов, то есть, не существует компьютера, у которого есть связь с самим собой через другие компьютеры.

Эмгыр сидит за компьютером с номером 1, а школьный учитель за компьютером с номером N . Эмгыр может взломать любой компьютер, с которым есть прямая связь. После взлома Эмгыр получает доступ к взломанному компьютеру и может взламывать еще и те, с которыми есть прямая связь от взломанного. На взлом компьютера с номером i , кроме первого, надо потратить t_i времени. В любой момент времени Эмгыр может взламывать не больше одного компьютера. Кроме того, в момент, когда начинается взлом компьютера i (кроме первого), с вероятностью p_i Эмгыр будет замечен преподавателем и у него останется D времени, пока ему не помешают.

Помогите Эмгыру определить вероятность взлома учительского компьютера, если Эмгыр будет действовать оптимально.

Входные данные

В первой строке задано три целых числа N ($2 \leq N \leq 10^6$) — количество компьютеров в сети, M ($1 \leq M \leq 2 \cdot 10^6$) — количество связей между компьютерами и D ($1 \leq D \leq 10^9$) — время, которое останется у Эмгыра на взлом сети после обнаружения.

Во второй строке заданы $N-1$ целых чисел — времена t_i ($1 \leq t_i \leq 1000$), необходимые на взлом компьютера i (где i от 2 до N).

В третьей строке заданы $N-1$ чисел — вероятности p_i ($0 \leq p_i \leq 1$) обнаружения перед взломом компьютера i (где i от 2 до N).

Затем идут M пар различных чисел U, V ($1 \leq U, V \leq N$), обозначающих, что из компьютера U есть связь с компьютером V (но не наоборот). Гарантируется, что заданная сеть ациклична.

Выходные данные

Выведите одно число — вероятность взлома учительского компьютера при оптимальной стратегии с точностью до 6 знаков после точки.

Пример

Входные данные	Результат
5 5 8 3 4 6 3 0.5 0.5 0.5 0 1 2 2 3 3 5 1 4 4 5	0.500000



Ф. Сигнализация

Ограничение на время работы: для C и C++ — 1 секунда, для Java и Kotlin — 9 секунд
Ограничение на использование памяти: 256МБ

Только что у Гены Трайкэча закончилась модульная контрольная по Экономике Разработки Улучшений Направленных На Дополнительную Автоматизацию (ЭРУННДА). Гена сразу же отправился на остановку, но на полпути он обнаружил, что так помогший ему на контрольной смартфон остался забытым на полочке под партой. Вернувшись в учебный корпус X-OSPU (eXtreme Open Source Programming University), Гена увидел, что аудитория уже закрыта и поставлена на сигнализацию. К счастью, в преподавательской Гене дали ключ, но, чтобы забрать смартфон, ему нужно было ещё разобраться с сигнализацией.

Любую аудиторию в X-OSPU можно представить в виде прямоугольного поля, разделённого на клетки. Часть клеток заняты партами или другой мебелью и поэтому непроходимы. По остальным клеткам можно двигаться вперёд, назад, влево и вправо. Вход в аудиторию может находиться в любой проходимой клетке, не обязательно у стены. В такие необычные аудитории студенты и преподаватели попадают по лестнице через люк в полу — не забывайте, что обозначает буква X в названии ВУЗа.

Сигнализации в аудиториях X-OSPU основаны на датчиках давления, установленных в некоторых клетках. Если зайти в такую клетку, запускается таймер, по истечении которого срабатывает сигнализация. В каждой клетке с датчиком находится свой независимый таймер со своим начальным значением. Чтобы отключить сигнализацию, нужно до истечения таймеров добраться до клетки, в которой находится пульт управления сигнализацией. Пульт может находиться в любой проходимой клетке аудитории.

Гена преодолевает одну клетку аудитории за одну секунду. Помогите ему найти такой путь от входа в аудиторию до пульта управления сигнализацией, чтобы двигаясь по нему Гена успел добраться до пульта до того, как срабатывает сигнализация. Найдите длину кратчайшего из таких путей либо сообщите Гене, что таких путей нет.

Входные данные

В первой строке даны числа W и H ($1 \leq W, H \leq 400$) — ширина и высота прямоугольного поля, описывающего аудиторию. Далее следует H строк по W символов «.» или «#», описывающих клетки аудитории. Символ «.» обозначает проходимую клетку, а символ «#» — непроходимую.

Затем в отдельной строке даны числа R_s и C_s — номера строки и столбца клетки поля, в которой находится вход в аудиторию. В следующей строке даны числа R_f и C_f — номера строки и столбца клетки, в которой находится пульт управления сигнализацией.

Затем в отдельной строке дано число B ($B \geq 0$) — количество клеток с датчиками. Далее следует B строк, описывающих датчики. Каждый датчик описывается тремя числами: R_i , C_i и T_i ($1 \leq T_i \leq 10^9$) — номерами строки и столбца клетки с датчиком и начальным количеством секунд на таймере этого датчика. Гарантируется, что датчики находятся только в проходимых клетках, и что в каждой клетке находится не более одного датчика.

Все номера строк и столбцов даны в диапазонах $1 \leq R_s, R_f, R_i \leq H$ и $1 \leq C_s, C_f, C_i \leq W$.

Выходные данные

Выведите одно целое число — длину искомого кратчайшего пути. Если добраться до пульта управления сигнализацией так, чтобы она не успела сработать (или вообще) невозможно, выведите число -1.



Пример

Входные данные	Результат
<pre> 5 5#.#. .#.#. .#.#. 1 5 5 5 2 3 3 4 3 5 1 </pre>	8
<pre> 4 3 .#.. .##. 1 1 1 3 1 3 3 3 </pre>	- 1

В первом примере, если отметить вход, пульт управления и датчики, то поле будет выглядеть следующим образом (поле приведено в таком формате только в целях пояснения):

```

.....S
.#.#.
.#4#1
.#.#.
.....F
                    
```

Оптимальным является путь через средний проход. В этом случае Гена успеет попасть в клетку с пультом в последний момент перед срабатыванием сигнализации. Если бы начальное время на таймере датчика в клетке (3,3) было 3 или меньше секунд, Гене пришлось бы идти через левый проход.

Во втором примере добраться от входа до пульта невозможно, так как единственный путь проходит через клетку с датчиком, и начального времени на таймере этого датчика не достаточно, чтобы добраться до пульта.



Г. Чеканной монетой

Ограничения: время работы — 1 секунда, использование памяти — 256МБ

Монтер Мечников был неумолим: «Утром деньги — вечером стулья», так что Остапу Бендеру с Кисой Воробьяниновым пришлось направиться к своему кормильцу — Пятигорскому провалу. Там охотники за стульями взимали с наивных граждан плату за вход, якобы для ремонта Провала.

Обаятельный и харизматичный великий комбинатор запрашивал у посетителей 29-копеечную монету, а отец русской демократии 10-копеечную. В сумме им нужно было собрать ровно N копеек, при этом потревожив минимальное количество граждан, из сугубо гуманистических соображений, конечно. Сколько раз предстояло совершить экспроприацию Остапу Бендеру (он же великий комбинатор), а сколько Кисе Воробьянинову (он же отец русской демократии)?

Входные данные

В единственной строке входа указано целое положительное число N ($1 \leq N \leq 10^{18}$) — число копеек, в которые монтер Мечников оценил казенный стул с возможно зашитыми в него бриллиантами.

Выходные данные

Выведите в единственной строке через пробел целое неотрицательное число A — сколько 29-копеечных монет должен собрать Остап, и целое неотрицательное число B — сколько 10-копеечных монет должен собрать Киса.

Если собрать нужную сумму монетами таких номиналов невозможно, выведите `Impossible`.

Пример

Входные данные	Результат
11	Impossible
175	5 3



Н. Илмюзия прогреста

Ограничение на время работы: для C и C++ — 1 секунда, для Java и Kotlin — 4 секунды
Ограничение на использование памяти: 256МБ

Однажды Влад и Саша решили придумать новый генератор случайных имен для переменных, чтобы упростить процесс разработки приложений (выбор имен занимает уйму времени). Но спустя день активного использования своего приложения они пришли к мысли, что имена получаются слишком скучные. Чтобы исправить эту ситуацию, они решили генерировать более интересные имена, а именно названия, в которых отсутствуют две подряд одинаковые буквы. Но поскольку ребята ленивые, то переписывать саму генерацию они не стали, а вместо этого просто меняют какие-то буквы на другие, чтобы имя стало интересным. Увы, алгоритм у них написан настолько плохо, что теперь приходится очень долго ждать, пока генератор придумает очередное имя.

Ребята просят вас помочь с этим. А именно: написать программу, которая по имеющемуся имени S сделает его интересным. Имя интересное, если в нем не встречаются две подряд одинаковые буквы. Буквы в названии являются маленькими латинскими литерами. За буквой «z» в алфавите снова следует буква «a» (циклический алфавит).

Учтите, что ребята просят менять буквы не на случайные, а на следующие по алфавиту. Менять одну букву можно несколько раз. За каждую такую замену начисляется одно штрафное очко. Разумеется, вам нужно набрать их как можно меньше.

Входные данные

В единственной строке входа указана строка S длиной $1 \leq \text{len}(S) \leq 10^5$. Строка состоит из маленьких латинских литер.

Выходные данные

Выведите измененную строку согласно правилам, предложенным Владом и Сашей. Не забывайте, что вам нужно набрать как можно меньше штрафных баллов. Если таких строк возможно больше одной (с одинаковым минимально возможным штрафом), то выведите минимальную лексикографически.

Пример

Входные данные	Результат
odessa	odesta
puzzle	puazle



I. Состав команд, или состязание равных

Ограничение на время работы: для C и C++ — 1 секунда, для Java и Kotlin — 2 секунды
Ограничение на использование памяти: 256МБ

Во время проведения некоторого турнира каждый из его участников, обозначенных номерами от 1 до N , приобрел рейтинг R_i . В завершение турнира его организаторы решили провести командное состязание, для чего составить из всех его участников две команды таким образом, чтобы в каждой команде количество людей было одно и то же и суммы рейтингов тоже совпадали.

Ваша задача — если такое разделение на команды возможно, вывести один из возможных вариантов, если невозможно — вывести число -1.

Входные данные

В первой строке дано одно четное число N ($2 \leq N \leq 30$). Во второй строке через пробел даны N целых неотрицательных чисел — рейтинги R_i ($0 \leq R_i \leq 60$) участников в порядке возрастания присвоенных номеров от 1 до N .

Выходные данные

Если разделение на команды по заданным условиям возможно, вывести две строки: в первой строке номера участников, вошедших в первую команду ($N/2$ чисел через пробел), а во второй строке номера участников, вошедших во вторую команду в том же формате.

Если разделение по заданным условиям невозможно, вывести в одной строке число -1.

Пример

Входные данные	Результат
6 11 3 5 7 12 2	2 3 5 1 4 6
6 3 2 6 1 2 4	1 5 6 2 3 4
10 1 3 5 7 9 11 13 15 17 19	-1

В первом примере в составе первой команды три участника турнира с номерами 2, 3 и 5; сумма рейтингов $R_2+R_3+R_5=3+5+12=20$. В составе второй команды — три участника турнира с номерами 1, 4 и 6; сумма рейтингов $R_1+R_4+R_6=20$.

Во втором примере приведен один из возможных вариантов разделения на команды. В третьем примере разделение по заданным условиям невозможно.



Ж. Гиперсемена

Ограничения: время работы — 1 секунда, использование памяти — 256МБ

Ринсвинд раздобыл на улице Алхимиков N гиперсемян и собирается все их рассадить по горшкам. Гиперсемена примечательны тем, что размер урожая гиперсемян с каждого горшка зависит и стремительно зависит от того, сколько их было посажено. Если в горшок посадить 1 гиперсемя, то 1 и вырастет, если посадить 2, вырастет уже 4, если 3 — 27, 4 — 256 и так далее, ну вы поняли. Однако Ринсвинду непременно нужно, чтобы суммарный урожай со всех горшков делился на 47. Сколько горшков он должен взять и сколько гиперсемян посадить в каждый из них? Вопрос, влезет ли урожай в занимаемую им Вселенную, Ринсвинда тревожит слабо.

Входные данные

В единственной строке задано целое число N ($1 \leq N \leq 74$).

Выходные данные

В первой строке выведите количество горшков M ($1 \leq M \leq N$). Во второй строке выведите M целых положительных чисел A_i , равных в сумме N . i -е число означает, что Ринсвинд посадит в i -й горшок A_i гиперсемян.

Если ни при какой рассадке получить урожай, кратный 47, нельзя, выведите единственную строку «Impossible» (без кавычек).

Если ответов несколько, выведите любой.

Пример

Входные данные	Результат
5	Impossible
11	4 1 4 2 4



К. Кубики

Обмеження на час роботи: для C та C++ — 1 секунда, для Java та Kotlin — 3 секунди
Обмеження на використання пам'яті: 256МБ

Маленькій бубочці Вові на день народження подарували кубики. Він побудував з цих кубиків n веж, вежа з номером i мала висоту в a_i кубиків. Але зовсім скоро ці вежі йому набридли і він вирішив перебудувати з уже наявних кубиків вежі так, щоб вежа з номером i мала висоту b_i . Процес перебудови виглядає так: Вова бере кубик з однієї побудованої вежі і перекладає його на іншу довільну вежу.

Скажіть, чи можливо перебудувати вежі так, як цього хоче Вова, і якщо так, то яку мінімальну кількість кубиків доведеться перемістити для цього?

Вхідні данні

Перший рядок містить єдине ціле число n ($1 \leq n \leq 10^5$) — кількість веж. Другий рядок містить n чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$), розділених пробілом — початкові висоти веж. Третій рядок містить n чисел b_1, b_2, \dots, b_n ($1 \leq b_i \leq 10^9$), розділених пробілом — висоти веж, які необхідно отримати.

Вихідні данні

Виведіть єдине число — мінімальну кількість кубиків, яку необхідно перемістити для перебудови, або -1, якщо це неможливо.

Приклад

Вхідні данні	Результат
5 5 3 2 3 1 1 1 7 3 2	6
3 1 2 3 1 2 2	-1