

Задача А. Боги

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

Во время очередного пира в Асгарде боги общались между собой. Под действием хмеля они конечно же поссорились. Все вело к драке, и когда она началась, боги осознали, что хотят драться только с теми, с кем повздорили, и нежелательно ссориться еще и с другими богами.

Один, как самый сообразительный бог Асгарда, сразу понял, что будет очень удобно, если боги смогут встать в две линии друг напротив друга таким образом, чтобы в одной линии не было повздоривших друг с другом. Тор решил доработать идею отца и ввел понятие «линия атаки». Линия атаки — это отрезок, соединяющий позиции поссорившихся богов. В своей доработке Тор предложил, что будет еще удобней, если линии атаки разных богов не будут пересекаться, так как тогда сами боги гарантированно не пересекутся во время боя. Все боги одобрили эту идею, осталось только занять позиции, и можно начинать драку.

Вам, как почетному жителю Мидгарда, посчастливилось помочь богам. Ваша задача написать программу, которая разобьет богов на две линии таким образом, чтобы выполнялась доработка Тора.

Формат входного файла

В самой первой строке заданы числа n ($1 \leq n \leq 10^5$) — количество богов в Асгарде. Далее следуют n строк, содержащих $k_i + 1$ число: число $k_i + 1$ ($1 \leq k_i \leq n$) — число богов, посорившихся с богом номер i , и их номера. Гарантируется, что если бог с номером i повздорил с богом с номером j , то бог с номером j повздорил с богом с номером i . Также гарантируется, что сумма по всем k_i не превосходит $4 \times n$.

Формат выходного файла

Если решение существует, выведите в первой строке два числа m и k — количество богов в первой и второй линии. В следующих двух строках выведите номера богов первой и второй линии в порядке их следования слева направо. Иначе выведите -1 .

Пример

<code>gods.in</code>	<code>gods.out</code>
4	2 2
1 2	1 3
2 1 3	2 4
2 2 4	
1 3	
3	-1
2 2 3	
2 3 1	
2 1 2	

Задача В. Javalhalla

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

Немногие знают, что уже викинги имели дело с языками программирования. К сожалению, у них не было компьютеров, чтобы их запускать. Однако, некоторые их программы дошли до наших дней, и у нас наконец есть возможность их запустить. Вам необходимо исполнить одну из таких программ.

В этой задаче мы опишем один из таких языков, дошедший до наших дней под именем «Javalhalla». В этом языке присутствуют два типа данных: целые числа и списки целых чисел. Все операции в этом языке производятся только со списками. Числа в списках нумеруются целыми числами, начиная с единицы. В этом языке присутствуют следующие операции:

- `List a = new List(x, y, ..., z)`. Создать новый список из чисел, перечисленных в скобках и присвоить ему имя *a*.
- `List b = a.subList(from, to)`. Взять все числа из списка *a*, находящиеся на позициях от *from* до *to* ($from \leq to$) и составить из них список *b*. Все изменения списка *a* отражаются в соответствующих позициях списка *b*, и наоборот. Обратите внимание на тестовые примеры и комментарии к ним.
- `a.set(i, x)` Присвоить *i* элементу списка *a* значение *x*
- `a.add(x)` Добавить число *x* в конец списка *a*. Эта операция применима только в том случае, если список *a* был создан так, как описано в первой операции (то есть список *a* не был создан, как подсписок какого-либо другого списка).
- `a.get(i)` Вывести *i* элемент списка *a* в отдельной строке

Формат входного файла

В первой строке находится целое число n ($1 \leq n \leq 100\,000$) — количество строк в программе. Каждая из следующих n строк описывает очередную команду в формате, описанном выше. Гарантируется, что никакой список не будет объявлен дважды, все обращения к спискам по индексу не выходят за границы списка. Все названия списков являются строками из строчных латинских букв длиной не более 10. Слова «new», «list», «set», «add» и «get» являются зарезервированными и не могут быть названием списка. Внутри скобок между аргументами и запятыми пробелов нет. Все числа во входном файле целые, положительные и не превосходят 10^9 . Суммарная длина списков, созданных с помощью первой операции, не превосходит 100 000.

Формат выходного файла

Для каждой операции `a.get(i)` выведите *i* элемент списка *a*.

Примеры

java.in	java.out
3 List a = new List(2,3,5) List b = a.subList(2,3) b.get(1)	3
5 List p = new List(2,4,8,16) p.get(4) List q = new List(3,9,27) q.add(5) q.get(4)	16 5
13 List x = new List(1,2,5,14,42) List y = x.subList(1,4) List z = y.subList(2,4) y.set(1,7) x.get(1) z.get(1) z.set(2,100) x.get(3) y.get(3) x.add(132) x.set(5,43) x.get(5) y.get(4)	7 2 100 100 43 14

Комментарий

Обратите внимание, что в третьем тестовом примере четвертая операция изменила как список *y*, так и список *x*, а седьмая — списки *x*, *y* и *z*.

Задача С. Молот Тора

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

В Мидгарде снова проблемы. И снова виноват во всем Локи. На этот раз он создал иллюзию и заменил во всем Мидгарде трехмерное пространство на двумерное. Тор, конечно же, пытался его остановить, но безуспешно. Но мало того, что Локи всё-таки успешно выполнил свой коварный план, так Тор еще и потерял где-то в Мидгарде свой молот Мьёльнир и теперь не может его найти.

Тор не очень хорошо знаком с математикой, но он догадался, что после того, как трехмерное пространство поменялось на двумерное, его молот из параллелепипеда превратился в прямоугольник. Он даже сумел посчитать площадь этого прямоугольника. Также до Тора дошла информация о местоположении двух противоположных углов его молота в Мидгарде. Теперь Тор хочет найти местоположение двух остальных углов. Но он не силен в математике и программировании, поэтому попросил Вас помочь ему. Помогите Тору: найдите хотя бы одно возможное расположение молота.

Поторопитесь, ведь иначе Тору не удастся остановить Локи, который уже начал строить еще более коварные планы по захвату и уничтожению Мидгарда!

Формат входного файла

В первой строке входного файла содержится одно вещественное число s ($0 < s \leq 10^9$) — площадь Мьёльнира в двумерном пространстве.

Две следующие строки входного файла содержат по два целых числа x_1, y_1, x_2, y_2 ($-10^4 \leq x_1, y_1, x_2, y_2 \leq 10^4$) — координаты противоположных углов молота.

Площадь дана не более, чем с шестью знаками после запятой.

Формат выходного файла

В двух строках выходного файла выведите пары координат двух точек молота, про которые Тор хочет узнать. Проверка равенства противоположных сторон выведенного вами четырехугольника, равенства его диагоналей и равенства его площади заданной будет производиться с точностью 10^{-4} .

Если существует несколько ответов, выведите любой.

Если ответа не существует, выведите четыре нуля, по два в каждой строке.

Пример

<code>mjolnir.in</code>	<code>mjolnir.out</code>
2.000000 0 -1 0 1	-1.0 0.0 1.0 0.0
1.000000 0 0 1 1	0.0 1.0 1.0 0.0
1.000000 0 0 0 1	0 0 0 0

Задача D. Один

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

Рагнарек должен наступить с минуты на минуту. Но Один не знает, когда именно он наступит.

Локи, как самый умный, сказал Одину формулу, по которой можно вычислить время наступления Рагнарека. К сожалению, Одину с одним глазом сложно уследить за всеми скобками в выражении, поэтому он решил заменить скобки первого уровня на фигурные, а второго уровня — на квадратные, остальные же оставить круглыми. То есть выражение $x * (y + (z * (5 + 3)))$ изменится на $x * \{y + [z * (5 + 3)]\}$.

Но Один не умеет программировать, Один умеет драться, поэтому программу придется написать вам. Не гневите Одина.

Формат входного файла

В первой и единственной строке входного файла указана формула, состоящая из букв латинского алфавита, знаков арифметических действий, пробелов, цифр и скобок. Гарантируется, что скобки образуют правильную скобочную последовательность. Длина формулы не превышает 10^5 .

Формат выходного файла

Выведите эту строку в формате удобном для чтения Одину.

Пример

<code>odin.in</code>	<code>odin.out</code>
<code>x * (y + (z * (5 + 3)))</code>	<code>x * {y + [z * (5 + 3)]}</code>
<code>1 * (0 + k + i) * (t - h + (0 * r))</code>	<code>1 * {0 + k + i} * {t - h + [0 * r]}</code>

Задача Е. Покер

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

Локи — очень коварный и хитрый бог. Но и у него бывают промашки: Тор догадался, кто остриг наголо его жену, и теперь требует от него компенсации. И Локи предложил ему такую сделку: в финале по асгардскому покеру он будет пасовать при каждой возможности, и таким образом Тор выиграет.

Каждый раунд в асгардском покере выкладывают на стол три карты, а затем после каждой фазы кладут на стол еще по одной карте. Всего фаз четыре. Перед каждым раундом игрок ставит либо малый блайнд, либо большой блайнд, который в два раза больше малого. После каждого раунда тип блайнда у игроков меняется на противоположный. Если денег у игрока меньше, чем размер блайнда, он ставит их все. После каждой фазы у игрока есть возможность удвоить ставку, если у него есть достаточное количество денег, либо отказаться дальше играть. Если какой-либо игрок спасовал, все деньги, которые были поставлены, переходят к другому игроку; игрок, который спасовал, теряет все свои поставленные деньги. В конце раунда игроки вскрывают карты. Рука составляется из любых пяти карт игрока и карт на столе. Игрок, чья рука сильнее, побеждает и забирает все деньги, которые были поставлены в этом раунде.

Локи очень скучно смотреть, как Тор выигрывает, поэтому ему очень хочется знать, сколько раундов всего будет в игре.

Формат входного файла

В первой строке даны числа x, b ($1 \leq x, b \leq 10^9$) — количество денег у Локи перед началом игры и размер малого блайнда. Во второй строке находится буква «L» или «T», если первый малый блайнд у Локи или Тора соответственно. Количество денег у Тора можно считать достаточным для любой ставки.

Формат выходного файла

В единственной строке выходного файла выведите количество раундов в этой игре.

Пример

<code>poker.in</code>	<code>poker.out</code>
7 2 L	3
7 2 T	3

Задача F. День Рагнарёк

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

Вот и настал день Рагнарёк. Чудовищный волк Фенрир проглотил Солнце, погрузив тем самым мир во тьму, из глубин морей всплыл мировой змей Ёрмунгланд. К ним примкнул и огненный великан Сурт с пылающим мечом, и повелительница загробного царства Хель, и коварный бог огня Локи вместе с инеистыми великанами — хримтусами и ётунами. Из Хельхейма приплыл корабль мертвецов Нагльфар, а войско сына Муспельхейма проскакало по мосту Биврёст, при этом разрушив его.

Асгарда, Мидгарда, Йотунхейма и Муспельхейма больше нет, нет и страны гномов, и страны эльфов. Но высоко-высоко в небе, там, где раньше был Асгард, но только еще выше его, стояли младшие боги. Это были молчаливый Видара, храбрый Вали, могучие Магни и Модри и меткий Ульр — они остались живы. Вместе с ними Бальдр и Ход, которым удалось вырваться из царства Хель. За поясом у Магни молот его знаменитого отца. Молодые боги разговаривают друг с другом, вспоминая дела и подвиги минувших веков, и строят для себя новую страну, а под ними из мирового моря опять подымается земля. Она вся зеленая, она покрыта чудесными лесами, садами, пастбищами и нивами.

Двое людей пережили Рагнарёк. Огонь Сурта не коснулся их. Они были далеко в глухом лесу, придумывали как остановить этот ужас, а когда придумали, мир вновь стал зелен и прекрасен. Питались они утренними росами. Это были женщина и мужчина, Лив и Ливтрасир. Они вышли из леса и народили детей, которые, тоже народив детей, заселили землю.

Для того, чтобы остановить Рагнарёк, Лив и Литрасир должны хором произнести волшебное слово. Волшебное слово должно одинаково читаться как справа налево, так и слева направо. А также, когда Лив и Литрасир произносят волшебное слово, они должны произнести имя малоизвестного бога, способного остановить Рагнарёк. При этом они хотят, чтобы длина волшебного слова была как можно меньше.

Формат входного файла

Во входном файле дано имя бога, способного остановить Рагнарёк. Длина имени не превосходит $3 \cdot 10^5$ и состоит из строчных латинских букв.

Формат выходного файла

Выведите минимальное по длине волшебное слово t , способное остановить Рагнарёк. Если таких слов несколько, ты выведите любое.

Примеры

<code>ragnarok.in</code>	<code>ragnarok.out</code>
<code>ab</code>	<code>aba</code>
<code>a</code>	<code>a</code>

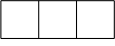
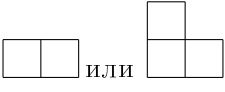

Задача G. Вальгалла

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 5 секунд
Ограничение по памяти: 256 мегабайт

Это интерактивная задача.

В Вальгалле очередной пир! Все эйнхерии сели за стол и ждут основного блюда. Основным компонентом этого блюда является мясо Сехримнира. Однако, на приготовление мяса требуется время. Каждый раз, когда очередная порция считается приготовленной, ее кладут в специальную тарелку, откуда пирующие могут взять себе три куска.

При этом, всё проходит по следующим правилам:

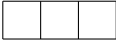


- ширина тарелки равна трем кускам мяса
- изначально тарелка пуста
- можно забрать только три куска мяса, лежащие горизонтально рядом;
 - после этого всё, что лежало выше забранных кусков, опускается на один слой вниз
- как только становится возможно взять три куска мяса, один из воинов его тут же забирает
- очередная приготовленная порция мяса может иметь одну из следующих форм: ,  или . Воины могут только выбирать то, как они положат это на тарелку. При этом, куски, относящиеся к одной порции, должны с охранять свое положение друг относительно друга в момент, когда порция кладется в тарелку.

Помогите воинам выбирать то, как следует класть на тарелку приготовленное мясо, чтобы на ней никогда не лежало больше 15 кусков (иначе голодные воины начнут несанкционированную драку и разнесут все чертоги).

Протокол взаимодействия с программой жюри:

Считайте число n ($1 \leq n \leq 50\,000$) — количество порций мяса. Затем, n раз сделайте следующее:

1. Считайте t ($t \in \{1, 2, 3\}$).

- $t = 1$ соответствует фигуре .
- $t = 2$ соответствует фигуре .
- $t = 3$ соответствует фигуре .

2. Выведите числа r ($0 \leq r < 4$) и c — сколько раз по часовой стрелке нужно повернуть фигуру и в какой столбец тарелки пойдёт самый левый столбец фигуры. c должно быть таким, чтобы ничего не оказалось вне тарелки. Столбцы в тарелке нумеруются с единицы.

Ваши действия должны быть такими, чтобы ни в какой момент времени на тарелке не было больше 15 кусков мяса. Можно считать, что воины забирают три куска, лежащие в одном ряду, моментально.

Пример

stdin	stdout
10	0 1
1	1 1
1	1 2
1	1 3
1	2 1
1	0 1
3	2 2
3	0 1
3	2 2
3	0 1
3	

Комментарий

Обратите внимание на то, что порции кладутся в тарелку сверху вниз — то есть каждый раз, когда мы кладем в тарелку новую порцию, ни над одним из ее кусков не должно находиться других кусков мяса.

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 Н. Вёлунд

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

По одной из версий «Песни о Вёлунде», Вёланд женился на лебединой деве Хервёр Чудесной, которая родила ему сына Хейме, а затем покинула его, оставив ему свое кольцо в качестве напоминания о себе. Однако, согласно той же версии Вёланд выковал также несколько дубликатов этого кольца. Оставив неотвеченным вопрос о цели этого действия, рассмотрим более внимательно сам этот процесс.

Известно, что у Вёланда было ровно m кузниц, n кузнецов и k дней на то, чтобы ковать эти кольца. При этом, каждый кузнец был готов работать только несколько (возможно, все или ноль) дней с номерами не меньшими, чем l_i , и не большими, чем r_i . Один кузнец успевал выковать за день работы ровно одно кольцо и требовал за день своей работы c_i золотых монет. При этом, естественно, он занимал в этот день одну из кузниц, в которой в этот день не мог работать никто, кроме него.

Вёланд хотел так распределить кузницы и рабочие дни, чтобы сделать как можно больше колец, а из всех таких вариантов выбрать тот, при котором он потратит меньше всего золота. Вам необходимо вычислить его результат и затраты.

Формат входного файла

Первая строка входного файла содержит три целых числа n , m и k ($1 \leq m \leq 100\,000$, $1 \leq n \leq 200\,000$, $1 \leq k \leq 10^8$) — количество кузнецов, кузниц и дней соответственно. Следующие n строк содержат описания кузнецов.

Каждая строка с описанием кузнеца содержит три числа l_i , r_i и c_i ($1 \leq l_i \leq r_i \leq k$, $1 \leq c_i \leq 10^4$) — номер первого и последнего из дней, в которые этот кузнец готов работать, и требуемое им количество монет за один день работы.

Формат выходного файла

Выведите два числа, разделенные пробелом — количество колец, которые удастся заполучить Вёланду, и минимальное количество монет, которые ему потребуется на это потратить.

Пример

<code>weyland.in</code>	<code>weyland.out</code>
3 2 5	9 15
1 5 1	
1 4 3	
2 3 2	

Комментарий

В приведенном примере первый кузнец будет работать все пять дней, второй — только в первый и в четвертый день, а третий — только во второй и в третий.