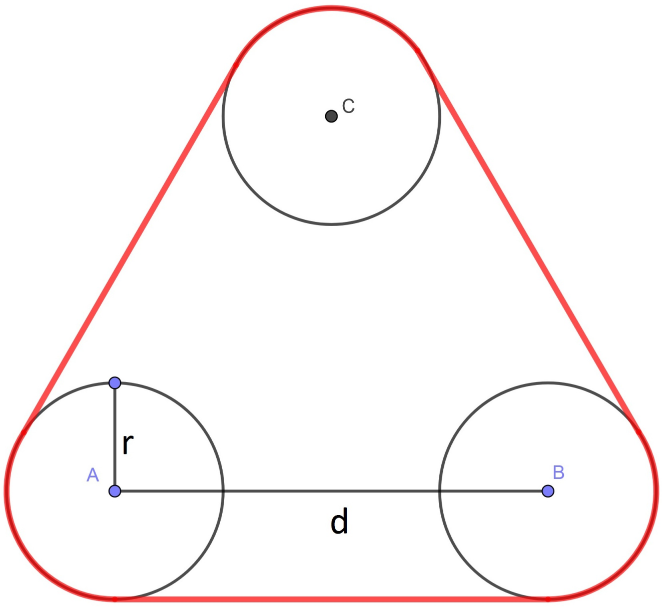
Задача A. Ремень ГУР

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

Гидроусилитель руля (ГУР) у автомобиля Айсиэна перестал работать: ремень порвался. Этот ремень огибает три шкива и передает вращение от ведущего к двум ведомым, один из которых и вращает ГУР. Эти шкивы представляют собой окружности, расположенные на равном удалении друг от друга — расстояние между любыми двумя равно d см (как показано на рисунке). Радиус каждого шкива равен r см.

Айсиэн, к сожалению, не знает какой длины ремень покупать, т.к. он купил машину совсем недавно. Помогите ему определить длину необходимого ремня.



Формат входных данных

В единственной строке заданы два целых числа, разделенных пробелом: d и r ($0 < d \leqslant 1000$, $2 \cdot r \leqslant d$).

Формат выходных данных

Вывести длину необходимого ремня с точностью до двух знаков после запятой.

Система оценки

В этой задаче нет подзадач. Каждый тест оценивается независимо от других.

Пример

стандартный ввод	стандартный вывод
`30 10`	`152.83`

Замечание

При расчетах число π следует принять равным 3.14159265.

Задача B. Инопланетная встреча

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

На далекой планете в системе Тау Кита разумные существа имеют по две руки, на каждой из которых по 11 пальцев, поэтому они используют 22-ичную систему счисления. Их система счисления также является позиционной (как и привычная нам десятеричная), но в ней используются следующие цифры: $0, 1, 2, \ldots, 9, A, B, C, \ldots, J, K, L$.

Во время недавней встречи с таукитянами представитель землян Эрчим задумался: каковы же максимальное и минимальное n-значные таукитянские числа, которые делятся на 2024, и все цифры которых не превышают заданного m. Помогите Эрчиму определить эти числа.

Формат входных данных

В единственной строке заданы два числа: целое десятеричное число n и цифра m в 22-ичной системе счисления ($2 \leqslant n \leqslant 10^6$, $m > 0$). Числа разделены пробелом.

Формат выходных данных

Выходные данные должны состоять из двух строк. В первую строку вывести искомое максимальное n-значное число в 22-ичной системе счисления. Во вторую строку — соответственно, искомое минимальное число.

Если решения нет, то вывести число -1.

Система оценки

Баллы за каждый тест в каждой подзадаче начисляются независимо от других тестов и подзадач. Ограничений на зависимость подзадач друг от друга нет.

№	**Дополнительные ограничения**	**Баллы**	**Информация о проверке**
1	$n \leqslant 10$	20	полная
2	$11 \leqslant n \leqslant 1010$	30	полная
3	$n \geqslant 1011$	50	полная

Примеры

стандартный ввод	стандартный вывод
`4 L`	`LIK0` `1320`
`5 9`	`99880` `10120`

Задача C. Преобразователь энергии

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

Земляне обустроили на Марсе колонию. Поверхность Марса представляет собой плоскость. На Марс постоянно падают метеориты. Земляне научились управлять метеоритами и изменять силу, с которой они упадут на поверхность Марса. Сила падения метеорита — это радиус ударной волны, вызванной данным событием. Атмосфера Марса стала очень плотной, поэтому метеориты изначально падают с силой 0, то есть без ударной волны.

Человечество столкнулось с энергетическим кризисом на Марсе, поэтому они построили n преобразователей энергии, которые вырабатывают 1 единицу энергии, если они окажутся в радиусе ударной волны метеорита. На поверхности Марса в некоторых точках расположены n преобразователей энергии.

Также было предсказано m точек, где произойдут m ближайших падений метеоритов. В каждую из точек упадёт ровно один метеорит.

Для прибора, изменяющего силу метеоритов, имеется запас в s батареек. Потратив r батареек, прибор может увеличить радиус ударной волны одного метеорита на r. Обратите внимание, что r может быть только натуральным числом. Каждую батарейку можно использовать только один раз.

Вам даны координаты точек, где находятся преобразователи энергии, и координаты мест ближайших m падений метеоритов. Выясните, какое максимальное количество единиц энергии земляне могут получить, используя прибор для изменения радиусов ударных волн метеоритов.

Формат входных данных

В первой строке заданы два целых числа n и s — количество преобразователей энергии и количество батареек соответственно ($1 \leqslant n \leqslant 10^4$, $0 \leqslant s \leqslant 1000$).

Следующие n строк содержат по два целых числа x_i, y_i, разделенных пробелом, — координаты точек, где располагаются преобразователи энергии ($-10^9 \leqslant x_i, y_i \leqslant 10^9$).

В следующей строке задано целое число m — количество метеоритов, которые упадут на поверхность Марса ($1 \leqslant m \leqslant 100$).

Следующие m строк содержат по два целых числа u_i, v_i, разделенных пробелом, — координаты точек, куда упадут метеориты ($-10^9 \leqslant u_i, v_i \leqslant 10^9$).

Гарантируется, что все $n + m$ точек попарно различны.

Формат выходных данных

Выведите одно число — максимальное количество единиц энергии, которое могут выработать преобразователи энергии.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены. Для некоторых подзадач может также требоваться, чтобы были пройдены все тесты из условия. Для таких подзадач дополнительно указана буква У.

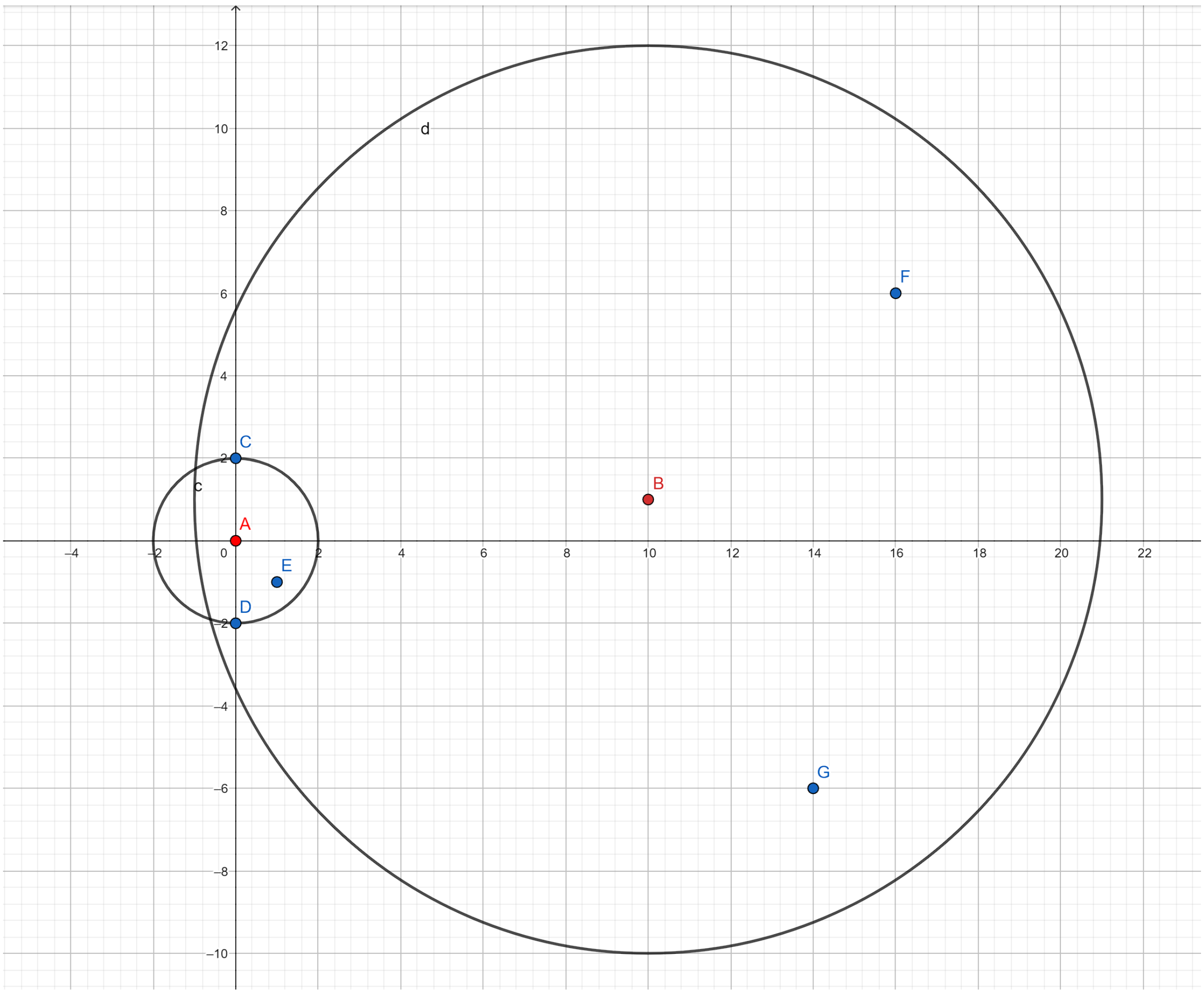
№	Дополнительные ограничения	Баллы	Необходимые подзадачи	Информация о проверке
1	$m = 2$	7	—	первая ошибка
2	$m \leqslant 3$, $s \leqslant 100$	8	—	первая ошибка
3	$n = 1$	12	—	первая ошибка
4	$n = 2$, $m \leqslant 18$	13	—	первая ошибка
5	Дополнительных ограничений нет	60	У, 1, 2, 3, 4	первая ошибка

Пример

стандартный ввод	стандартный вывод
5 13 0 2 0 -2 1 -1 16 6 14 -6 2 0 0 10 1	8

Замечание

Рисунок к примеру:



A и B — это точки падения метеоритов. Остальные точки — это места расположения преобразователей энергии. Для одного из способов получения максимального количества единиц энергии, первому метеориту земляне устанавливают радиус ударной волны равным 2, и он накрывает 3 преобразователя, каждый из которых даст 1 единицу энергии. Второму метеориту изменяют силу на 11, и он накрывает все 5 преобразователей. В итоге будет выработано 8 единиц энергии.

Задача D. MultiChad

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

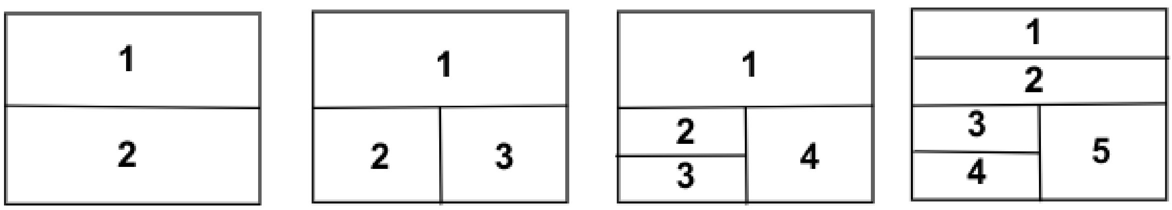
Коля разрабатывает новый терминальный мультиплексер — «MultiChad». Исходно терминал — это квадратное окно $[0, 2^k] \times [0, 2^k]$.

Пока Коля разработал поддержку только четырех команд. Первые две из них реализуют функции деления окна терминала на подокна:

- Если поступает команда `split v i`, то окно номер `i` разбивается вертикальной линией на два одинаковых подокна. При этом левое подокно сохраняет свой номер `i`, а правое получает номер `i+1`. Соответственно, нумерация всех остальных окон, начиная с `i+1`-го, увеличивается на 1. Также в дальнейшем, когда окна будут разбиваться вертикально, номера в левой половине всегда будут меньше номеров в правой половине.

- Если поступает команда `split h i`, то окно номер `i` разбивается горизонтальной линией на два одинаковых подокна. При этом верхняя половина сохраняет свой номер `i`, а нижняя половина получает номер `i+1`. Аналогично, нумерация всех остальных окон, начиная с `i+1`-го, увеличивается на 1. Также в дальнейшем, когда окна будут разбиваться горизонтально, номера в верхней половине всегда будут меньше номеров в нижней половине.

Например, когда выполняется последовательность команд `split h 1`, `split v 2`, `split h 2`, `split h 1`, результат выглядит таким образом:



Также поддерживаются две следующие команды:

1. Команда `where i` — терминал сообщает координаты левого верхнего угла окна с номером `i`.

2. Команда `which y x` — терминал сообщает, окно с каким номером покрывает точку с координатами `(y, x)`. Если точка лежит ровно на границе окна, то считаем окно покрывающим её. Если точка покрыта двумя или более окнами, то терминал сообщает наименьший из их номеров.

Левая верхняя точка терминала — это точка с координатами $(y = 0,\ x = 0)$, ось y направлена вниз, ось x направлена вправо.

Формат входных данных

В первой строке заданы два целых числа: q — количество команд и k (2^k — длина стороны терминала) $(0 < q \leqslant 3 \cdot 10^5,\ 0 < k \leqslant 62)$.

В следующих q строках заданы команды в вышеописанном формате. Гарантируется, что все команды корректны (команды типа `split` и `where` принимают номер существующего на тот момент окна, `split h` не делит окно высоты 1, `split v` не делит окно ширины 1, `which` принимает целые координаты) $(0 \leqslant y, x \leqslant 2^k)$.

Формат выходных данных

Для каждой команды `where i` выведите в отдельной строке два числа y и x, разделённые пробелом, — координаты левого верхнего угла окна i.

Для каждой команды `which y x` выведите в отдельной строке одно число i — номер окна, содержащего точку (y, x).

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены. Для некоторых подзадач может также требоваться, чтобы были пройдены все тесты из условия. Для таких подзадач дополнительно указана буква У.

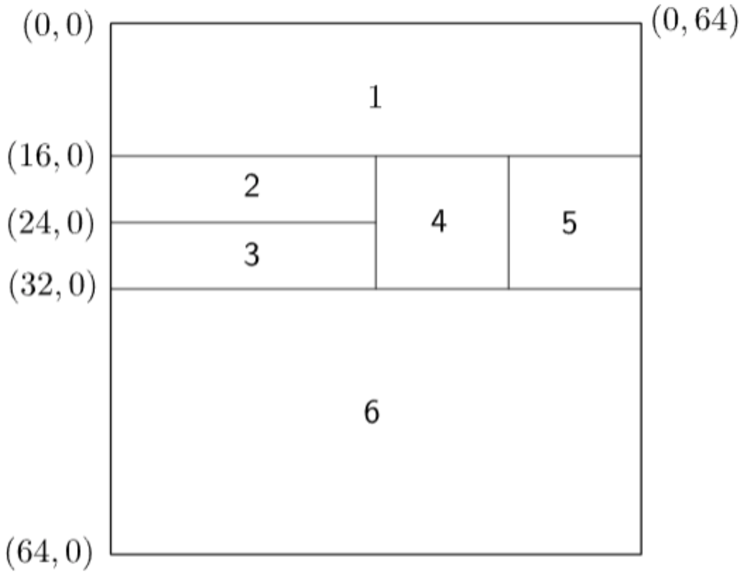
№	Дополнительные ограничения	Баллы	Необходимые подзадачи	Информация о проверке
1	$q \leqslant 140$, только команды `split v 1` и `where i`	10		первая ошибка
2	$q \leqslant 140$, только команды `split v 1`, `where i` и `which y x`	10	1	первая ошибка
3	Только команды `split v i` и `where i`	10	1	первая ошибка
4	Только команды `split v i`, `where i` и `which y x`	10	1, 2, 3	первая ошибка
5	Только команды `split v i`, `split h i` и `where i`	20	1, 2, 3	первая ошибка
6	Нет дополнительных ограничений	40	У, 1, 2, 3, 4, 5	первая ошибка

Примеры

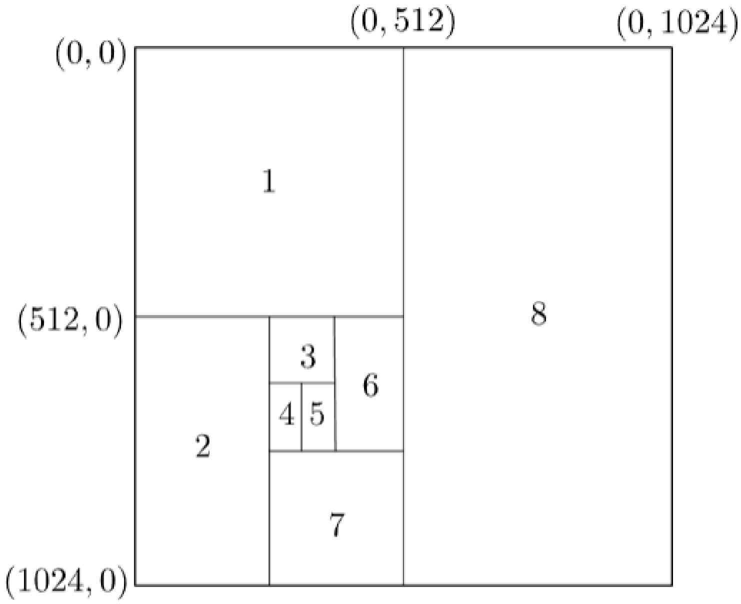
стандартный ввод	стандартный вывод
13 6 where 1 split h 1 split h 1 where 1 split v 2 split v 3 which 30 48 where 3 split h 2 where 5 where 6 which 24 5 which 10 60	0 0 0 0 3 16 32 16 48 32 0 2 1
11 10 split v 1 split h 1 split v 2 which 400 256 split h 3 split v 3 where 6 split h 3 split v 4 which 256 768 where 5	1 0 512 8 640 320

Замечание

В первом примере результирующий терминал выглядит так:



Во втором примере результирующий терминал выглядит так:



Задача E. Светлое будущее

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	256 мегабайт

Союз X состоит из N бурно развивающихся стран. В 0-й год был разработан план, согласно которому каждый год выбирается страна с номером A (у каждой страны имеется свой уникальный номер), в которую будет инвестировано 3^k монет, где k — номер года. Чтобы жителям остальных стран не было обидно, в страны, в которые было вложено суммарно не больше, чем в страну A, также предусматривается инвестиция в размере 3^k монет. При этом учитываются вложенные ранее 0-го года денежные суммы.

Страны могут как вступать в союз X, так и выходить из него. При вступлении страны считается, что в неё было вложено 0 монет, даже если она ранее уже состояла в данном союзе. Считается, что страны вступают в союз и покидают его до выбора страны для инвестирования.

По имеющимся данным за K лет, включающим номера выбранных стран и информацию о вступлении и выходе стран из союза X, выведите список стран союза через K лет в порядке невозрастания суммарного объема инвестиций.

Формат входных данных

В первой строке заданы два целых числа N и K ($1 \leqslant N, K \leqslant 10^5$).

В следующих N строках содержатся номер страны A_i и вложенная в неё ранее сумма I_i ($0 \leqslant A_i \leqslant 10^9$, $0 \leqslant I_i \leqslant 10^9$).

Затем в следующих K строках заданы события в k-й год: количество J_k стран, вступивших в союз и их номера, количество L_k вышедших из союза стран и их номера, а также номер A_k страны, которая была выбрана для инвестиции.

Гарантируется, что сумма J_k и L_k не превосходит 10^5. Также страна не может выйти и войти в союз в один и тот же год. Все числа в одной строке разделены пробелом.

Формат выходных данных

В первой строке выведите количество стран через K лет.

В следующей строке выведите номера стран союза в порядке невозрастания суммарного объема инвестиций через K лет. Если есть несколько возможных ответов — выведите любой из них.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены. Для некоторых подзадач может также требоваться, чтобы были пройдены все тесты из условия. Для таких подзадач дополнительно указана буква У.

№	Дополнительные ограничения	Баллы	Необходимые подзадачи	Информация о проверке
1	$1 \leqslant K \leqslant 35$	10	У	первая ошибка
2	$N, K, \sum J_k, \sum L_k \leqslant 10^3$	10	У	первая ошибка
3	$N, K, \sum J_k, \sum L_k \leqslant 10^4$	40	У, 1, 2	первая ошибка
4	Дополнительных ограничений нет	40	У, 1, 2, 3	первая ошибка

Пример

стандартный ввод	стандартный вывод
3 3 1 4 2 10 3 9 1 4 0 3 0 0 1 0 1 4 1	3 1 3 2

Замечание

В данном примере в 0-й год сначала к союзу X присоединяется страна с номером 4. Затем в страны 1, 3 и 4 инвестируется по $3^0 = 1$ монете.

К началу 1-го года страны имеют следующие суммарные инвестиции: в страну 1 — 5, 2 — 10, 3 — 10 и 4 — 1. В этот год в страны 1 и 4 инвестируется по $3^1 = 3$ монеты.

К началу 2-го года в страну 1 будет вложено 8 монет, в 2 — 10, в 3 — 10 и в 4 — 4. В этот год страна 4 покидает союз, поэтому $3^2 = 9$ монет будет вложено только в страну 1.

В итоге через 3 года в страну 1 вложено 17 монет, в 2 и 3 — по 10 монет.

Задача F. Телепортация

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	256 мегабайт

Школьник Вася очень часто опаздывает в школу, и это стало такой большой проблемой, что он изобрёл у себя в гараже телепортатор.

Район проживания Васи можно представить как прямоугольную матрицу, длина которой равна M, а высота — N. В данном районе имеются его дом и школа. Клетки в матрице бывают двух типов:

1. `.` — тротуар или иные места, по которым Вася может ходить. Скажем, что такие места *безопасные*.
2. `#` — лужи или дороги, по которым Вася не может ходить. Скажем, что такие места *опасные*.

Вася свободно может перемещаться на одну клетку вверх, вниз, влево и вправо пешком, если там находится безопасное место. То есть, если он находится в клетке (i, j), то он может перейти в клетку $(i+1, j)$, $(i-1, j)$, $(i, j+1)$ или $(i, j-1)$, если эти клетки безопасные и не выходят за края матрицы. Это действие не тратит никаких ресурсов.

При первом же испытании Вася осознает, что сделал ошибку в разработке телепортатора. Оказывается, он телепортирует не туда, куда захочет Вася, а в такую точку, которая высчитывается следующим образом:

- В телепортаторе имеется генератор: $F(x, m) = (ax + c) \mod m$.
- Если Вася находится в координатах (i, j), то при применении телепортатора он окажется в точке с координатами $(F(i, N), F(j, M))$.

Важно отметить, что Вася не хочет оказаться в опасной зоне, поэтому телепортацию нужно проводить только тогда, когда он уверен, что окажется в безопасном месте. Телепортация доступна в любой момент из любой безопасной точки.

Телепортатор имеет K батареек. При каждой телепортации разряжается 1 батарейка

Ваша задача — определить, сможет ли Вася добраться до школы, и если да, то сколько батареек будет израсходовано. Нужно определить минимальное количество израсходованных батареек.

Формат входных данных

- Первая строка содержит натуральные числа N, M — размеры матрицы, высота и длина соответственно ($N, M \leqslant 10^3$).
- Следующие N строк содержат по M символов (либо `"."`, либо `"#"`), описывающих матрицу.
- В следующей строке заданы два целых числа — координаты начальной позиции Васи (i_0, j_0) $(0 \leqslant i_0 < N,\ 0 \leqslant j_0 < M)$.
- Затем идут два целых числа — координаты школы (i_s, j_s) $(0 \leqslant i_s < N,\ 0 \leqslant j_s < M)$.
- Далее идут три натуральных числа a, c, K — параметры генератора и количество батареек в телепортаторе соответственно ($a, c < 2^{32}$ и $K < 10^3$).

Формат выходных данных

Вывести минимальное количество израсходованных батареек, необходимых Васе для достижения школы. Если это невозможно, вывести -1.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены. Для некоторых подзадач может также требоваться, чтобы были пройдены все тесты из условия. Для таких подзадач дополнительно указана буква У.

№	**Дополнительные ограничения**	**Баллы**	**Необходимые подзадачи**	**Информация о проверке**
1	$N, M \leqslant 20$	20	У	первая ошибка
2	$N, M \leqslant 100$	30	У, 1	первая ошибка
3	Дополнительных ограничений нет	50	У, 1, 2	первая ошибка

Пример

стандартный ввод	стандартный вывод
6 4 #... .### .... .#.# .#.# #..# 0 1 2 0 7 3 14	1

Задача G. Инопланетный шифр

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

В противостоянии с инопланетянами в Якутии было создано партизанское движение. В ходе секретной миссии была перехвачена зашифрованная переписка инопланетян. Эта информация может быть крайне полезной для освобождения нашей планеты, поэтому её необходимо расшифровать. Наши шпионы выяснили способ, которым была зашифрована эта информация.

Как оказалось, инопланетяне очень сильны физически, но их наука еще не так продвинута, как на Земле, и они пока не освоили алгоритмы асимметричного шифрования. Инопланетное шифрование представляет собой следующее: инопланетяне берут секретную информацию, кодируют её в двоичной системе счисления, затем из получившейся двоичной строки H длины N составляют массив A, где A_i $(0 \leqslant i \leqslant N)$ — это длина самого длинного палиндрома строки H, который начинается с i-го символа. Собственно, перехваченная информация представляет собой массив A.

Ваша задача — из массива A восстановить строку H. Среди всех возможных строк, удовлетворяющих заданным условиям, необходимо найти лексикографически наименьшую.

Формат входных данных

В первой строке задано одно целое число N — длина строки H $(1 \leqslant N \leqslant 10^5)$

Вторая строка содержит N целых чисел $A_1, A_2, \ldots, A_N$ $(1 \leqslant A_i \leqslant N)$ — длины самых длинных палиндромов строки H, которые начинаются с i-го символа.

Формат выходных данных

В единственную строку выведите бинарную строку H без пробелов.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены. Для некоторых подзадач может также требоваться, чтобы были пройдены все тесты из условия. Для таких подзадач дополнительно указана буква У.

№	Дополнительные ограничения	Баллы	Необходимые подзадачи	Информация о проверке
1	$1 \leqslant N \leqslant 10$	10	У	первая ошибка
2	$1 \leqslant N \leqslant 10^4$	40	У, 1	первая ошибка
3	Дополнительных ограничений нет	50	У, 1, 2	первая ошибка

Пример

стандартный ввод	стандартный вывод
4 3 3 1 1	0101

Задача H. Деление факториалов

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

Тимур очень любит делить числа. Сегодня он в школе узнал, что такое факториал натурального числа: факториалом $k!$ числа k называется число $1 \cdot 2 \cdot 3 \cdot \ldots \cdot (k-1) \cdot k$. С новыми знаниями Тимур придумал себе следующее занятие:

1. Сначала он выбирает некоторое простое число p.
2. Затем он строит следующую последовательность $\{a\}_n$: $a_0 = p$, $a_{i+1} = a_i + k$, где k — максимальная степень числа p, на которую делится факториал числа a_i при $i > 0$.

Тимуру стало интересно: при каком начальном простом p в последовательности встретится число x с наименьшим возможным номером?

Формат входных данных

В единственной строке задано одно натуральное число x ($2 \leqslant x \leqslant 4 \cdot 10^7$).

Формат выходных данных

Выведите два числа через пробел — наименьший возможный номер появления числа x в последовательности и соответствующее ему начальное простое число p. Можно показать, что ответ всегда существует при заданных ограничениях. Если существует несколько возможных ответов — выведите любой из них.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены. Для некоторых подзадач может также требоваться, чтобы были пройдены все тесты из условия. Для таких подзадач дополнительно указана буква У.

№	Дополнительные ограничения	Баллы	Необходимые подзадачи	Информация о проверке
1	$x \leqslant 100$	10	У	полная
2	$x \leqslant 1000$	10	У, 1	полная
3	$x \leqslant 10^4$	20	У, 1, 2	полная
4	$x \leqslant 10^6$	30	У, 1, 2, 3	полная
5	Нет дополнительных ограничений	30	У, 1, 2, 3, 4	полная

Примеры

стандартный ввод	стандартный вывод
4	1 3
7	0 7
10	3 7

Замечание

В первом примере можно взять простое число $p = 3$: тогда $a_0 = 3$, а $a_1 = 3 + 1$, так как $3! = 6$ делится только на 3^1.

В третьем примере число 10 можно получить из простого числа 7: факториал $7! = 5040$ делится только на 7^1, $8! = 40320$ делится только на 7^1, $9! = 362880$ делится только на 7^1.

Задача I. T ночей в СВФУ

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Талбан — бедный студент Северо-Восточного федерального университета, который только что слетел со стипендии. Ему пришлось устроиться охранником в своём университете, чтобы хоть как-то прокормить себя. К сожалению, днём у Талбана лекции, поэтому работать он может только в ночную смену, с 12 часов ночи до 6 утра. После T ночей отработки он получит свою первую зарплату в размере 12000 рублей!

Однако не всё так просто. По ночам в СВФУ бродят голодные монстры, которые съедят любого человека, оставшегося в здании. К счастью, в офисе Талбана есть лазеры, которые он может включить, чтобы не дать монстрам войти внутрь. Включенные лазеры тратят 1 энергию в секунду. Одна ночь длится S секунд (в СВФУ время идёт по-другому), то есть смена Талбана начинается в нулевую секунду, и заканчивается в S-ую.

В i-ю ночь на Талбана будет охотиться n_i монстров. Дайаана, ясновидящая подруга Талбана, предсказала, что j-й монстр придёт к его офису в L_j-ю секунду, и уйдёт в R_j-ю секунду. Всё это время Талбан должен держать лазеры включенными, тратя по 1 энергии в секунду. Если несколько монстров в один момент будут стоять рядом с офисом, то лазеры всё ещё будут тратить по 1 энергии. Кроме лазеров, у Талбана есть электрошокер, которым он может воспользоваться один раз за ночь. Когда он использует шокер, все монстры, стоящие возле офиса, немедленно уходят (и не возвращаются), и лазеры можно выключить на некоторое время. Электрошокер не тратит энергию.

Так как у Талбана имеется ограниченное количество энергии, он хочет знать для каждой ночи, какое минимальное количество энергии он должен потратить, чтобы выжить до конца смены.

Формат входных данных

В первой строке заданы три целых числа T, S и G — сколько ночей Талбан должен отработать, сколько секунд длится его смена и номер подзадачи соответственно ($1 \leqslant T \leqslant 10^5$, $1 \leqslant S \leqslant 10^9$, $0 \leqslant G \leqslant 6$, где $G = 0$ означает тесты из условия).

Далее следуют T описаний ночей.

Первая строка каждой ночи содержит единственное целое число n — количество монстров в эту ночь ($1 \leqslant n \leqslant 2 \cdot 10^5$).

Каждая из следующих n строк содержит по два целых числа L_j и R_j — отрезок времени, в течение которого j-й монстр будет стоять у офиса ($0 \leqslant L_j < R_j \leqslant S$).

Гарантируется, что сумма n по всем ночам не превосходит $2 \cdot 10^5$.

Формат выходных данных

Для каждой ночи выведите минимальное количество энергии, которое Талбан должен потратить, учитывая, что он может пользоваться электрошокером.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены. Для некоторых подзадач может также требоваться, чтобы были пройдены все тесты из условия. Для таких подзадач дополнительно указана буква У.

Обозначим сумму n по всем ночам как N.

№	Дополнительные ограничения	Баллы	Необходимые подзадачи	Информация о проверке
1	$N, S \leqslant 1000$	15	У	первая ошибка
2	$N \leqslant 1000$	15	У, 1	первая ошибка
3	$S = 10^9$, $L_1 = 5 \cdot 10^8$, $R_1 = S$, и $R_i \leqslant L_1$ для всех $i \geqslant 2$	15	—	первая ошибка
4	Ни в один момент времени у офиса не будет стоять больше одного монстра	5	—	первая ошибка
5	$S \leqslant 10^5$	25	У, 1	первая ошибка
6	Дополнительных ограничений нет	25	У, 1, 2, 3, 4, 5	первая ошибка

Пример

стандартный ввод	стандартный вывод
5 11 0 1 3 9 3 3 5 6 11 0 3 4 0 4 3 5 1 4 2 6 6 4 6 2 8 0 6 8 11 3 8 1 10 5 2 11 3 10 4 9 5 8 6 7	0 5 3 7 4

Замечание

В первую ночь Талбан должен использовать шокер в 3-ью секунду, как только первый монстр подойдёт к офису. Тогда Талбан вообще не воспользуется лазерами, и потратит 0 энергии.

Во вторую ночь Талбану выгодно включить лазеры в 0-ую секунду, и выключить их в 5-ую, чтобы первый и третий монстры его не съели, а затем использовать шокер на второго монстра в 6-ую секунду. Лазеры потратили 5 энергии.

В третью ночь Талбану выгодно включить лазеры в 0-ую секунду, и выключить их в 3-ью, и в эту же 3-ью секунду использовать шокер на всех монстров одновременно. Тем самым он прогонит их всех, и лазеры потратят всего 3 энергии.

В четвертую ночь Талбан должен включить лазеры в 0-ую секунду, и выключить их в 4-ую, и в эту же 4-ую секунду использовать шокер на всех монстров, кроме четвертого, так как четвертый монстр ещё не возле офиса. Далее Талбан должен заново включить лазеры в 8-ую секунду, и так просидеть до конца смены (до 11-ой секунды), чтобы четвертый монстр не съел его. Итого лазеры использовали $4 + 3 = 7$ энергии.

Задача J. Инопланетные налоги

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

В параллельной реальности Якутия состоит из n населённых пунктов, соединённых m двусторонними дорогами. Из каждого населённого пункта можно попасть в каждый, переходя по дорогам. У каждого населённого пункта изначально есть свой *бюджет* — количество денег, находящихся в казне.

Всё в республике было хорошо, пока инопланетяне не захватили всю Землю, в том числе и Якутию. Очень долго новое инопланетное правительство собирало налоги с бедных землян.

Конечно же, жителям республики новый расклад был не по душе, и они решили дать отпор. У каждого населённого пункта есть свой уровень *сопротивления* — насколько сильно местное население. *Сопротивления* всех населённых пунктов попарно различны.

Земляне смогли прогнать незваных гостей только спустя q дней. Каждый из этих q дней в Якутии происходило одно из двух событий:

- Событие типа 1: населённый пункт v увеличивает свой *бюджет* на x посредством продажи бриллиантов.
- Событие типа 2: инопланетяне отправляют своих агентов с силой c и планом p в город v, чтобы они собрали налоги. Агенты уменьшают *бюджет* всех населённых пунктов на p, до которых они могут дойти. Агенты могут перемещаться только по дорогам, так как иначе дикая фауна Якутии съест их живьём. Также они ни в коем случае не могут посещать населённые пункты с *сопротивлением* строго больше c, ведь там их попросту нейтрализуют. Если в населённом пункте *бюджет* меньше, чем p, то инопланетяне забирают оттуда все деньги, оставляя поселение с нулевым *бюджетом.* Гарантируется, что *сопротивление* стартового города v не больше c.

Теперь, после долгой войны с инопланетными захватчиками, необходимо посчитать убытки. Вам, как министру инопланетных дел, поручили для каждого события второго типа посчитать, сколько суммарно *бюджета* инопланетяне забрали в этот день.

Формат входных данных

Первая строка содержит три целых числа n, m и G — количество населённых пунктов, количество дорог и номер подзадачи соответственно ($1 \leqslant n \leqslant 3 \cdot 10^5$, $0 \leqslant m \leqslant 6 \cdot 10^5$, $0 \leqslant G \leqslant 9$, где $G = 0$ означает тесты из условия).

Вторая строка содержит n целых чисел a_i — начальный *бюджет* i-го населённого пункта ($0 \leqslant a_i \leqslant 10^9$).

Третья строка содержит n целых чисел b_i — уровень *сопротивления* i-го населённого пункта ($1 \leqslant b_i \leqslant n$, каждое число встречается ровно по одному разу).

Каждая из следующих m строк содержит два целых числа u_i и v_i — населённые пункты, которые соединяет i-я двусторонняя дорога. Гарантируется, что все населённые пункты связаны. Петли и кратные дороги допустимы ($1 \leqslant u_i, v_i \leqslant n$).

Следующая строка содержит целое число q ($1 \leqslant q \leqslant 3 \cdot 10^5$).

Далее, i-я из следующих q строк содержит число t_i — какое событие произошло в день i:

- Если $t_i = 1$, то после этого даны ещё два целых числа v_i и x_i — номер населённого пункта и на сколько увеличился его *бюджет* соответственно. Учтите, что после этого события *бюджет* населённого пункта может стать больше 10^9 ($1 \leqslant v_i \leqslant n$, $0 \leqslant x_i \leqslant 10^9$).
- Если $t_i = 2$, то после этого даны ещё три целых числа v_i, c_i и p_i — стартовый населённый пункт агентов, их сила и план соответственно ($1 \leqslant v_i \leqslant n$, $1 \leqslant c_i \leqslant n$, $0 \leqslant p_i \leqslant 10^9$).

Формат выходных данных

Для каждого события второго типа выведите одно целое число — сколько суммарно *бюджета* забрали инопланетяне в этот день.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены. Для некоторых подзадач может также требоваться, чтобы были пройдены все тесты из условия. Для таких подзадач дополнительно указана буква У.

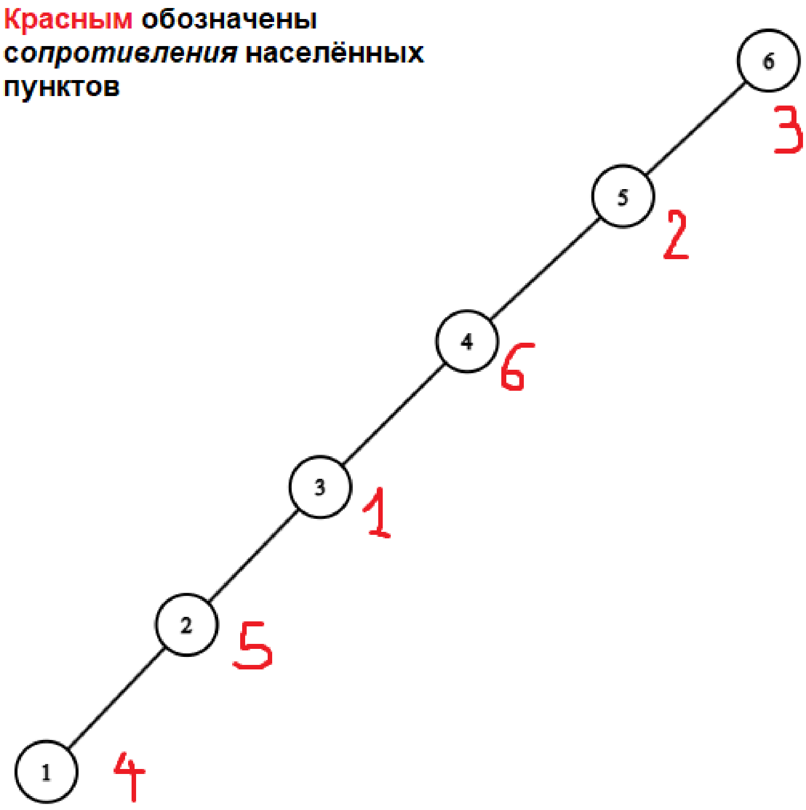
№	Дополнительные ограничения		Баллы	Необх. подзадачи	Информация о проверке
	n, m, q	Комментарий			
1	$n, q \leqslant 1000$, $m \leqslant 2000$	—	10	У	первая ошибка
2	—	Для всех событий второго типа $c_i = n$	10	—	первая ошибка
3	—	Для всех событий второго типа $v_i = 1$	10	2	первая ошибка
4	—	Гарантируется, что агенты не будут пытаться брать больше *бюджета*, чем есть в населённом пункте	10	—	первая ошибка
5	$n = m + 1$	Для всех дорог $u_i = i$ и $v_i = i + 1$	10	—	первая ошибка
6	$n = m + 1$	Для всех дорог $u_i = 1$ и $v_i = i + 1$	10	—	первая ошибка
7	$n = m + 1$	—	10	$5 - 6$	первая ошибка
8	$n, q \leqslant 10^5$, $m \leqslant 2 \cdot 10^5$	—	15	У, 1	первая ошибка
9	—	Дополнительных ограничений нет	15	У, $1 - 8$	первая ошибка

Примеры

стандартный ввод	стандартный вывод
6 5 0 10 12 17 12 13 9 4 5 1 6 2 3 1 2 2 3 3 4 4 5 5 6 7 2 1 5 11 1 2 17 1 6 8 2 5 2 10 2 2 6 15 1 5 6 2 6 5 3	32 10 51 5
5 6 0 20 20 20 20 50 1 2 3 4 5 1 3 3 4 4 2 2 5 5 1 1 2 4 2 3 3 10 2 1 4 20 1 5 50 2 2 5 99	30 50 99
1 0 0 15 1 3 1 1 20 2 1 1 5 2 1 1 40	5 30

Замечание

В первом примере Якутия выглядит следующим образом:



Начальные *бюджеты* населённых пунктов: $10, 12, 17, 12, 13, 9$.

В первый день инопланетяне посылают агентов с планом 11 и силой 5 в город 1. С силой 5 агенты могут добраться до городов 1, 2 и 3, так как *сопротивление* города 4 слишком большое. Агенты забирают у городов 2 и 3 по 11 *бюджета*, а у города 1 всего 10 *бюджета*, поэтому инопланетяне забирают всё, что есть. Итого агенты забрали $10 + 11 + 11 = 32$ *бюджета*.

После первого дня *бюджеты* населённых пунктов: 0, 1, 6, 12, 13, 9.

Во второй день город 2 продаёт бриллианты и увеличивает свой *бюджет* на 17.

После второго дня *бюджеты* населённых пунктов: 0, 18, 6, 12, 13, 9.

В третий день город 6 продаёт бриллианты и увеличивает свой *бюджет* на 8.

После третьего дня *бюджеты* населённых пунктов: 0, 18, 6, 12, 13, 17.

В четвёртый день инопланетяне посылают агентов с планом 10 и силой 2 в город 5. На этот раз агенты не могут покинуть свой стартовый город 5, так как *сопротивления* соседних городов больше их силы, поэтому они забирают 10 *бюджета* у одного города 5.

После четвёртого дня *бюджеты* населённых пунктов: 0, 18, 6, 12, 3, 17.

В пятый день агенты с силой 6 и планом 15 начинают в городе 2. Их сила позволяет посетить все населённые пункты, поэтому суммарно они заберут $0 + 15 + 6 + 12 + 3 + 15 = 51$ *бюджет*.

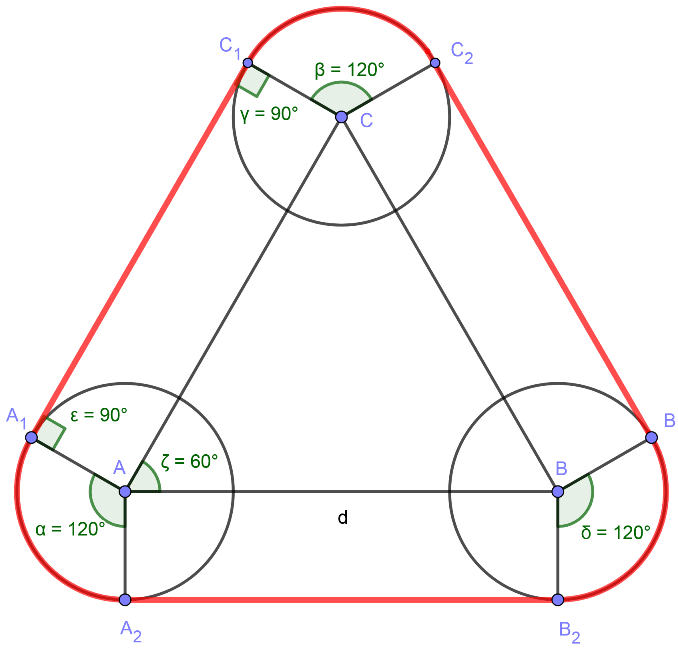
После пятого дня *бюджеты* населённых пунктов: 0, 3, 0, 0, 0, 2.

В шестой день город 5 увеличивает свой *бюджет* на 6.

После шестого дня *бюджеты* населённых пунктов: 0, 3, 0, 0, 6, 2.

В последний день агенты с силой 5 и планом 3 начинают в городе 6. Они могут добраться до городов 5 и 6, и суммарно забирают $3 + 2 = 5$ *бюджета*.

Задача A. Ремень ГУР



Ремень огибает все три шкива, следовательно, часть ремня, соединяющая любые два шкива, является касательной к ним. Опустим радиусы к точками касания. Без потери общности, рассмотрим четырёхугольник AA_1C_1C. Тогда углы $\angle AA_1C_1 = \angle CC_1A_1 = 90°$, а т.к. $AA_1 = CC_1 = r$, то $AC = A_1C_1 = d$. Аналогичным образом, $C_2B_1 = A_2B_2 = d$.

Остается определить длину сегментов A_1A_2, B_1B_2, C_1C_2. Заметим, что $\angle A_1AA_2 = 360° - 60° - 90° - 90° = 120°$. Аналогично, $\angle B_1BB_2 = \angle C_1CC_2 = 120°$, следовательно, сумма длин сегментов A_1A_2, B_1B_2, C_1C_2 равна длине окружности диаметром r, т.е. $2\pi r$.

Тогда длину всего ремня можно вычислить по формуле $3d + 2\pi r$, что и должна вывести ваша программа.

Задача B. Инопланетная встреча

Для решения этой задачи можно использовать несколько различных алгоритмов.

Пока $n < 7$, можно использовать полный перебор вариантов. Однако это неэффективный метод. Поскольку требуется находить числа, которые делятся на 2024, то быстрее будет, проверив очередное k, которое делится на 2024, сразу переходить к числу $k + 2024$. Таким образом можно проверить числа до длины $n < 11$.

Однако, чтобы решить эту задачу на полный балл, нужно заметить, что $2024 = 8 \cdot 11 \cdot 23$ и затем использовать признаки делимости на 8, 11 и 23.

Признак делимости числа на 8: число, состоящее из трёх последних цифр, должно делиться на 8. Этот признак работает в 22-ичной системе счисления так же, как и в десятеричной (доказательство основывается на том, что основания обеих систем счисления делятся на 2).

Признак делимости на 11 в 22-ичной системе счисления: последняя цифра в числе должна быть 0 или B (11). Данный признак работает аналогично признаку делимости на 5 в десятеричной системе счисления. Но учитывая, что искомые числа также должны делиться на 8, получается, что последняя цифра не может быть B. То есть использование данного признака приводит к тому, что у всех искомых чисел последняя цифра обязательно должна быть равна 0.

Признак делимости на 23 в 22-ичной системе счисления: разность сумм цифр, стоящих на чётных и нечётных местах в n-значном числе, должна делиться на 23. Этот признак аналогичен признаку делимости на 11 в десятеричной системе счисления (для доказательства нужно попробовать умножить число на 23 столбиком).

Далее можно заметить, что в записи максимального искомого числа левые $n - 5$ цифр будут одинаковыми. Конкретную цифру можно легко подобрать перебором. Оставшиеся 5 цифр справа в записи нужно подобрать так, чтобы выполнялись признаки делимости.

Искомое минимальное число всегда будет иметь 1 в первой позиции и $n-6$ нулей. Оставшиеся 5 цифр справа в записи нужно подобрать так, чтобы выполнялись признаки делимости.

Задача C. Преобразователь энергии

Для решения первой подзадачи достаточно перебрать радиус первого метеорита. Затем посчитать, сколько энергии вырабатывается в каждом из возможных вариантов.

Для решения второй подзадачи тоже можно использовать полный перебор. Надо перебрать r_1 и r_2, радиусы первого и второго метеоритов соответственно, а радиус третьего будет $s - r_1 - r_2$.

В третьей подзадаче каждый метеорит может накрыть максимум один преобразователь. Найдем для каждого метеорита минимальный радиус, при котором его ударная волна дойдет до преобразователя. Затем используем жадный алгоритм и будем изменять силу тех метеоритов, которым нужно меньше всего батареек, чтобы задеть преобразователь.

В четвертой подзадаче будем перебирать битовые маски длина m. Если i-й бит маски равен 1, то i-й метеорит должен накрыть два преобразователя. Иначе накрывает либо 0, либо 1 преобразователь. Затем мы вычитаем из s количество батареек потраченное на метеориты с единичным битом в маске. Получается у нас есть s_1 батареек и некоторое количество метеоритов с нулевым битом в маске. Для максимизации ответа сведем задачу к третьей подзадаче и используем жадный алгоритм.

Для полного решения воспользуемся динамическим программированием. Посчитаем $dp[i][j]$ — максимальное количество вырабатываемой энергии при использовании только первых i метеоритов и j батареек. В цикле будем перебирать i от 1 до m. Отсортируем точки по расстоянию от i-го метеорита. Теперь с помощью двух указателей можем посчитать для всех r от 1 до s $cnt[r]$ — количество преобразователей, которых накроет i-й метеорит, если ему установить радиус равным r. Зная, что $dp[i][j] = \max(dp[i-1][j], \max_{r=1\ldots j}(dp[i-1][j-r] + cnt[r]))$, посчитаем $dp[i][j]$ для всех j. Ответ будет максимумом среди $dp[m][j]$ по всем j.

Задача D. MultiChad

1. В первых двух подзадачах достаточно только следить за количеством выполненных команд `split v 1` — это число однозначно задаёт вид всего терминала. Это можно делать разными способами. Например, можно при каждой команде `where i` или `which y x` непосредственно посчитать координаты границ всех окон и легко ответить на запрос.

2. Для решения следующих двух подзадач для поддержки команд `split v i` можно воспользоваться PBDS-структурой `ordered_set` (либо можно непосредственно реализовать декартово дерево, но не рекомендуется). При выполнении любой команды `split` или `where`, нужно сначала определить местоположение окна с номером `i` — если хранить в ordered-set левые x границы всех окон, то окну i соответствует i-й элемент множества. Тогда на команду `where` мы отвечаем сразу, а на команду `split v i` надо создать новый элемент для нового окна (правая половина текущего окна i) и добавить его во множество. Для команды `which y x` можно бинарным поиском по x найти наибольший номер i такой, что его левая граница не превышает x — это и есть ответ.

3. Для решения последних двух подзадач следует посмотреть на задачу другим взглядом. Представим терминал как бинарное дерево — любое окно терминала соответствует вершине дерева. Когда окно делится на два подокна — у вершины появляются два сына. Также для каждой вершины будем хранить координаты его левого верхнего угла и его размеры, а также количество листьев в его поддереве. Это позволяет находить вершину i спуском по дереву — если слева в дереве находится не меньше i, то спускаемся влево, иначе вправо. На `where i` можем ответить немедленно, а на команду `split` мы создаем две новые вершины и подвешиваем их к текущей вершине, посчитав их координаты и размеры (зависит от типа `split`). Для команды `which y x` мы также делаем спуск по дереву — в нелистовой вершине мы знаем как расположены окна-дети и можем вычислить, в какой из них находится точка (y, x) и спуститься в него.

Задача E. Светлое будущее

Подзадача 1

Заметим, что число $a_i + \sum\limits_{k=0}^{34} 3^k \leqslant 10^9 + \frac{3^{35}-1}{2} < 2^{64}$ помещается в 64-битный целочисленный тип данных. Если каждый год считать обновления количества монет за $O(n)$, добавление и удаление стран за $O(n)$ и в конце отсортировать массив, получим асимптотику $O(kn + n \log n)$.

Подзадача 2

Данную подзадачу все еще можно решить, считая обновления как количества монет, так и стран, каждый год за $O(n)$. Но так как 3^{1000} не помещается ни в какой примитивный тип данных, придется использовать длинную арифметику.

Подзадача 3

Сначала посчитаем первые 20 лет как в подзадаче 1 и отсортируем города в порядке убывания. Далее мы будем хранить страны в порядке невозрастания инвестиций. Заметим, что $3^{20} > 10^9 + \sum\limits_{k=0}^{19} 3^k$, поэтому начиная с 20-го года, страны, в которые были сделаны инвестиции, с конца списка просто будут переходить в начало списка, и у нас нет необходимости явно считать инвестированные монеты. Для этого можно использовать обычный массив или дек, перенося страны каждый год за $O(n)$. Добавление и удаление стран можно считать за $O(n)$. Итоговая асимптотика $O(20n + n \log n + kn)$.

Подзадача 4

Нужно улучшить асимптотику обновлений после 20-го года. Для номеров стран заведем ассоциативный массив, который умеет вставлять/удалять/находить элемент за $O(\log n)$. Будем хранить в связном списке в отсортированном порядке количество инвестированных монет, а не номера стран. Чтобы быстро находить по стране его позицию в связном списке, будем хранить в ассоциативном массиве указатель на элемент связного списка (указатель на элемент связного списка остается валидным вплоть до его удаления). И так как мы можем быстро найти место для среза в связном списке, за $O(1)$ перенесем в начало его хвост из стран, в которые были инвестированы монеты. Итоговая асимптотика $O(20n + n \log n + k \log n)$. Если использовать хеш-таблицу, можно добиться усредненной асимптотики $O(20n + n \log n + k)$.

Задача F. Телепортация

Сначала требуется инициализировать заданный граф. Представим точки, то есть безопасные координаты, как вершины. Пройдемся по всем вершинам и найдем все смежные им вершины. Такими вершинами могут быть соседние на 1 клетку, если они безопасные, а также вершина, в которую Вася может телепортироваться, если она безопасная. Если между двумя вершинами имеется ребро, его вес может быть равен либо 0 (в случае, если в пешей доступности), либо 1 (если используем телепорт).

Для решения первой подзадачи можно использовать алгоритм Флойда-Уоршелла и затем определить расстояние между начальной позицией Васи и школой. Если n — количество вершин графа, то асимптотика $O(n^3)$.

Для второй подзадачи можно использовать алгоритм Дейкстры на заданном графе, используя цикл в цикле. Асимптотика $O(n^2)$.

Для полного решения нужно использовать алгоритм Дейкстры на двоичной куче. Для него асимптотика $O(n \log n)$. Но также заметим, что можем здесь использовать алгоритм `0-1 BFS`, поскольку ребра имеют вес либо 0, либо 1. Для него асимптотика $O(n)$.

Задача G. Инопланетный шифр

Давайте зададим несколько утверждений, которые помогут нам добраться до решения:

- Поскольку подстрока, начинающаяся с i-го символа, не может быть длиннее $N-i+1$ символов, можно предположить, что $1 \leqslant A_i \leqslant N-i+1$ для любого i ($1 \leqslant i \leqslant N$).

- Для любого $i < N$, если i-й и $(i+1)$-й символы одинаковы, то всегда существует палиндром, начинающийся с i-го символа, длина которого не менее 2 символов. Поэтому A_i всегда больше 1, если $H_i = H_{i+1}$. Также можно заключить, что если $A_i = 1$, то $H_i \neq H_{i+1}$.

Получив эти утверждения, мы можем восстановить строку H.

Для решения этой задачи нам нужно начать с последнего символа. Примем его значение за 1.

$$\underbrace{\_\,\_\,\_\cdots\_\,\_\,\_1}_{N \text{ characters}}$$

Для каждого j-го символа, начиная с $N-1$ и заканчивая 1: если $A_j = 1$, то H_j равен противоположному $H_j + 1$, иначе $H_j = H_{j+A_j-1}$.

Применив этот алгоритм, мы получим некоторую двоичную строку H, которая будет иметь соответствующий набор символов для соответствия массиву A. Но в утверждении стоит задача найти *лексикографически минимальный* ответ. Чтобы убедиться в этом, мы можем проверить первый символ вычисляемой строки H. Если первый символ H_1 равен 1, то мы можем просто инвертировать всю строку, чтобы получить строку H, которая будет начинаться с 0, таким образом, мы получим лексикографически меньшую строку. Полученная строка и будет нашим ответом.

Задача H. Деление факториалов

Заметим, что любые два соседних простых числа отличаются не более чем в два раза (подобные факты для заданных ограничений можно было проверить, написав решето Эратосфена, или любым другим образом проверяя числа на простоту из необходимого диапазона), и что число $k!$ делится на p только 1 раз, если $k < 2p$.

Из этого следует, что самое большое простое число p, не превосходящее данное число x, задает последовательность, где у числа x будет индекс $i = x - p$. Также можно не рассматривать простые числа из диапазона $\left(\frac{x}{2}; p\right)$, так как в них число x будет иметь индекс ровно $i = x - q$ и $q < p$. Аналогично можно не рассматривать простые числа больше чем $x - p$, так как числа в последовательности будут расти более чем $x - p$ раз ровно на 1. Так как $x - p < 210$ в заданных ограничениях, то можно проверить простые числа только до 210.

Чтобы быстро вычислять элементы последовательности, можно воспользоваться формулой Лежандра (она легко выводится): степень вхождения простого числа p в факториал $k!$ равно $v_p(k!) = \left\lfloor \frac{k}{p} \right\rfloor + \left\lfloor \frac{k}{p^2} \right\rfloor + \left\lfloor \frac{k}{p^3} \right\rfloor + \ldots.$

Задача I. Т ночей в СВФУ

Переформулируем задачу. Нам дано n отрезков. Мы можем выбрать одну точку, и у всех отрезков, которые содержат выбранную точку, правая граница сдвигается к этой точке. Надо найти минимальную возможную длину объединения этих отрезков.

Подзадача 1

В этой подзадаче можно было перебрать все целые моменты времени, в которые мы могли использовать электрошокер. Далее мы сдвигаем правые границы затронутых отрезков и за $O(S)$ считаем ответ. Асимптотика такого решения $O(S(S+n))$.

Подзадача 2

Заметим, что нам невыгодно использовать электрошокер, когда в выбранный момент времени не появляется новых монстров. Действительно, ведь мы можем воспользоваться им на секунду раньше, и ответ не ухудшится. Тогда мы можем перебрать n левых границ отрезков и использовать шокер только в них. Далее мы сдвигаем правые границы затронутых отрезков, и нам осталось посчитать их длину объединения. Эта известная задача решается сканлайном за $O(n \log n)$ (логарифм появляется от сортировки событий). Итоговая асимптотика $O(n^2 \log n)$.

Подзадача 3

В этой подзадаче нам было выгодно использовать электрошокер на первого монстра, чтобы сэкономить целых $5 \cdot 10^8$ единиц энергии. Все остальные монстры атакуют только в первую половину смены, и на них больше не осталось заряда электрошокера. Значит, задача свелась к обычному объединению отрезков. Асимптотика $O(n \log n)$.

Подзадача 4

В этой подзадаче нам было выгодно использовать электрошокер на монстра, который стоит у офиса дольше всех. Асимптотика $O(n)$.

Подзадача 5

Давайте заведём дерево отрезков с массовыми операциями на S элементов, где i-ое значение равно количеству монстров, стоящих у офиса в i-ую секунду (учтите, что правые границы отрезков не учитываются, так как в эти моменты монстры уходят). Изначально прибавим 1 на каждом отрезке, ведь мы пока не использовали электрошокер.

Решим задачу сканлайном. У нас будут события двух типов: 1) монстр приходит, и 2) монстр уходит. Отсортируем их в порядке возрастания координаты (в случае равенства сначала должны идти события второго типа). Когда мы рассматриваем событие первого типа, мы скажем, что использовали шокер в этот момент времени. Тогда отнимем 1 на рассматриваем отрезке, так как этот монстр моментально ушёл. Также нам надо прибавить на отрезке от прошлого события до текущего события значение, равное количеству монстров на этом отрезке (это значение можно легко поддерживать), ведь мы передвинули использование шокера, и все монстры на этом отрезке вернулись. Когда мы рассматриваем событие второго типа, делаем всё то же самое, но не отнимаем на текущем отрезке 1, ведь монстр ушёл сам по себе.

Теперь нам нужно узнать ответ для текущего события. Ответом является количество ненулевых элементов на всём массиве. Для поддержки этого значения в вершине ДО будем хранить минимум и количество минимумов. Ответом является S минус количество минимумов на всём отрезке, ведь минимум всегда равен нулю. В итоге у нас получилось решение за $O(n \log S + S)$.

Подзадача 6

Чтобы получить полный балл по этой задаче, надо ещё сжать координаты, и получить решение за $O(n \log n)$.

Задача J. Инопланетные налоги

Подзадача 1

Чтобы решить эту подзадачу, достаточно с помощью обхода графа в глубину (DFS) находить все достижимые вершины и отнимать у них бюджет. На каждый запрос второго типа мы ответим за $O(n)$. Итоговая асимптотика $O(nq)$.

Подзадача 4

Так как гарантируется, что агенты не будут забирать больше бюджета, чем есть в вершине, нас вообще не интересуют запросы первого типа. Для каждого запроса второго типа мы должны просто найти количество достижимых вершин, чтобы вычислить ответ. Давайте отвечать на запросы в оффлайне. Для каждой возможной силы агентов сохраним все стартовые вершины. Пусть в графе у нас изначально нет вершин. Заведём СНМ, в котором будем поддерживать размеры компонент связностей. Пройдёмся по всем возможным уровням сопротивления от 1 до n. Пусть мы сейчас рассматриваем сопротивление x. Добавим в граф вершину с этим сопротивлением и с помощью СНМ объединим её со всем соседями. Затем рассмотрим все стартовые вершины агентов силы x и узнаем размеры компонент связностей, в которых эти вершины находятся. СНМ работает за $O(n\alpha(n))$. Ответы на запросы мы находим за $O(1)$. Итоговая асимптотика $O(n\alpha(n) + q)$.

Подзадача 2

В этой подзадаче агенты могли добраться до всех вершин. Получается, нам надо поддерживать 3 типа запросов: прибавлять в точке, отнимать на всём массиве и для каждого элемента a_i делать $a_i = max(a_i, 0)$. Со всем этим справляется структура данных Segment Tree Beats. Однако, существует решение без STB, использующее обычное дерево отрезков. Давайте скажем, что вершина обнулилась, если её бюджет стал равен нулю. Тогда заметим, что суммарно вершины перейдут из ненулевого состояния в нулевое не более, чем $n + q$ раз. Действительно, обнуленная вершина может снова стать ненулевой после запроса первого типа, но запросов первого типа не больше q. Тогда давайте в вершине ДО хранить количество ненулевых вершин, их суммарный бюджет, минимальный бюджет, и вершину с минимальным бюджетом.

Теперь перед тем, как уменьшать бюджеты, узнаем количество ненулевых вершин cnt. Текущим ответом будет $cnt \times p$. Далее уменьшим бюджеты всех вершин. После этой операции какие-то вершины могли обнулиться. Рассмотрим вершину с минимальным бюджетом, если её бюджет неположительный, отнимаем от текущего ответа лишний бюджет, помечаем эту вершину как обнуленную, и прибавляем к ней $+1e18$ бюджета, чтобы мы её повторно не рассматривали. Повторяем этот процесс, пока минимум не станет положительным. В конце у нас будет корректный ответ. Итоговая асимптотика $O((n + q) \log n)$.

Подзадача 3

То же самое, что и во второй подзадаче, но заранее сортируем все вершины в порядке увеличения минимального возможного сопротивления от 1 вершины. Теперь все запросы у нас на префиксе. Асимптотика $O((n + q) \log n + m)$.

Подзадача 5

Бинарным поиском + разреженной таблицей для каждого запроса второго типа найдём, до какой вершины слева и справа агенты могут дойти. Теперь воспользуемся деревом отрезков из второй подзадачи. Асимптотика $O((n + q) \log n)$.

Подзадача 6

Отсортируем вершины в порядке возрастания сопротивления. Воспользуемся деревом отрезков из второй подзадачи. Для каждого запроса второго типа проверим, можем ли мы попасть в 1 вершину. Если нет, то уменьшаем бюджет только стартовой вершины. Иначе будем отнимать бюджеты у префикса вершин. Асимптотика $O((n + q) \log n)$.

Подзадачи 7-9

В этих подзадачах надо было для каждой вершины понять, до каких вершин агенты могут дойти, если их сила равна сопротивлению этой вершины. Будем строить дерево. Заведём СНМ как в четвертой подзадаче, но вместо размеров будем для каждой компоненты связности хранить вершину с максимальным сопротивлением. Пройдёмся по вершинам в порядке возрастания сопротивления. Для текущей вершины рассмотрим всех соседей. Если сопротивление соседа больше нашего, или он находится в нашем множестве, то игнорируем его. Иначе найдём вершину с максимальным сопротивлением из множества соседа, и проведём ребро в дереве между нами и этой максимальной вершиной. В конце у нас получится дерево с корнем в вершине с сопротивлением n. Заметим, что если агенты начнут в какой-то вершине с силой, равной сопротивлению этой вершины, то они смогут дойти до всех вершин в поддереве, и только до них. Однако сила агентов не обязательно равна сопротивлению стартовой вершины. В этом случае можно бинарными подъемами подняться до самой высокой вершины, чьё сопротивление не больше силы агентов. С помощью эйлерового обхода сведём задачу на дереве к задаче на массиве. Далее воспользуемся деревом отрезков из второй подзадачи. Итоговая асимптотика $O((n + q) \log n + m)$. В зависимости от оптимальности написанного решения, получаем от 85 до 100 баллов.