

Задача Armstrong. Билеты на поезд

Имя входного файла:	input.txt или стандартный поток ввода
Имя выходного файла:	output.txt или стандартный поток вывода
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Вот-вот стартует заключительный этап Открытой олимпиады школьников, и преподавателям из других городов пора задуматься об отправлении детей к месту проведения олимпиады. Конечно же, легче всего добраться до Москвы на поезде, но для этого надо озаботиться покупкой билетов. Именно с этой задачей сегодня столкнулся Денис — ответственный за олимпиады по программированию в замечательном городе Кислозаводске. Внимательно изучив расписание поездов, он пришел к выводу, что удобнее всего будет добираться на фирменном поезде «Капкан», ходящим напрямую из Кислозаводска в Москву. К сожалению, данный состав ходит не чаще двух раз в неделю, поэтому Денис выбрал конкретный рейс, на котором поедут все школьники.

Так исторически сложилось, что все вагоны поезда имеют разную вместимость. Кроме того, вагоны отличаются условиями комфорта, поэтому цены на билет в разные вагоны тоже отличаются. Все места внутри одного вагона стоят одинаково.

По результатам отборочного этапа n учеников из данного города прошли на заключительный этап, и естественно, Денис хочет потратить как можно меньше денег на билеты. Сначала эта задача показалась ему очень простой, но потом Денис вспомнил про введенный недавно закон, гласящий, что детей в столь длительных поездках должны сопровождать взрослые. Кроме того, закон строго регламентирует необходимое количество взрослых для присмотра за группой детей, на каждые k школьников нужен хотя бы один взрослый сопровождающий. Стоит отметить, что взрослый может присматривать только за детьми из того же вагона, что и он сам. Более формально, если в вагоне едут x детей, то в этом же вагоне должны находиться хотя бы $\lceil \frac{x}{k} \rceil$ взрослых. При этом Денис может уговорить на участие в поездке сколь угодно много сопровождающих, так что по этому параметру ограничений нет. Разумеется, за билеты для сопровождающих тоже нужно платить.

Теперь перед Денисом стоит куда более сложная задача, помогите ему справиться с ней как можно быстрее: отправить всех детей и необходимое количество взрослых на олимпиаду данным поездом, не нарушая при этом закон и потратив минимально возможное количество денег. Если невозможно отправить всех детей и нужное количество взрослых данным поездом, то выведите «-1». В противном случае выведите ответ — минимальную суммарную стоимость всех билетов.

Формат входных данных

В первой строке входных данных записаны три целых числа m , n и k ($1 \leq m \leq 10^6$, $0 \leq n \leq 10^9$, $1 \leq k \leq 20$) — количество вагонов в поезде, количество детей, прошедших на заключительный этап, и максимальное количество детей на одного сопровождающего.

В следующих m строках записаны по два целых числа p_i и c_i ($1 \leq p_i, c_i \leq 10^9$) — количество свободных мест и стоимость одного места в i -м вагоне.

Формат выходных данных

Выведите одно число — минимальную суммарную стоимость билетов, которую необходимо потратить для отправления всех школьников, или «-1», если всех детей отправить невозможно.

Примеры

ВВОД	ВЫВОД
5 12 3 10 7 11 6 6 2 7 10 8 5	70
8 55 1 70 22 69 9 99 44 31 81 57 50 22 79 99 33 17 64	1536

Пояснение

Обратите внимание, что билет Денису покупать не нужно, он предпочитает путешествовать на велосипеде.

В первом тесте из условия оптимальным способом будет отправить 4 детей с 2 взрослыми в третьем вагоне, 6 детей с 2 взрослыми в пятом вагоне и 2 детей с 1 взрослым во втором вагоне. Таким образом, в каждом вагоне на каждых трёх детей будет хотя бы один взрослый, а общая стоимость поездки составит $6 \cdot 2 + 8 \cdot 5 + 3 \cdot 6 = 12 + 40 + 18 = 70$.

Система оценки

Тесты к этой задаче состоят из четырёх групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Дополнительные ограничения		Комментарий
		n	m	
0	0	–	–	Тесты из условия
1	19	$n \leq 100$	$m \leq 100$	
2	30	–	$m \leq 1000$	
3	28	–	$m \leq 10\,000$	
4	23	–	–	Offline-проверка

Задача Magellan. Выбор вагона

Имя входного файла:	input.txt или стандартный поток ввода
Имя выходного файла:	output.txt или стандартный поток вывода
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Вася любит ездить на поездах, но не любит, когда вагон, в котором он едет, расположен в хвосте.

Вася садится на поезд на вокзале. Поезд состоит из n вагонов, пронумерованных от 1 до n , считая от локомотива. Во время движения поезда происходит три типа событий:

1. К голове поезда подцепляют некоторое количество вагонов;
2. К хвосту поезда подцепляют некоторое количество вагонов;
3. Вася пересчитывает величину, с помощью которой он оценивает удобство вагона (подробнее о ней ниже).

В каждый момент времени будем нумеровать вагоны начиная с головы поезда, начиная с 1. При добавлении новых вагонов к голове поезда, нумерация старых может сдвинуться.

Чтобы выбрать, в каком вагоне ехать, Вася ввёл для каждого вагона величину A_i (где i — номер вагона), которая вычисляется следующим образом:

- В начале поездки $A_i = 0$, а также для новых вагонов, в момент их добавления.
- Во время очередного пересчёта Вася выбирает целые **положительные** числа b и s и прибавляет к A_i величину $b + (i - 1) \cdot s$.

Вася ещё не решил, откуда и куда ехать, поэтому после каждого события одного из трёх типов он хочет знать номер наиболее близкого к голове поезда вагона, что его значение A_i минимально. Так как вагонов очень много Вася попросил Вас написать программу, отвечающую на его вопрос.

Формат входных данных

Первая строка содержит два целых числа n и m ($1 \leq n \leq 10^9$, $1 \leq m \leq 300\,000$) — количество вагонов в поезде на момент отправления с вокзала и количество станций соответственно.

Следующие m строк задают описания событий. Каждое событие одного из трёх следующих видов:

- «1 k » ($1 \leq k \leq 10^9$), нужно подцепить к голове поезда k вагонов.
- «2 k » ($1 \leq k \leq 10^9$), нужно подцепить к хвосту поезда k вагонов.
- «3 b s » ($1 \leq b, s \leq 10^9$), нужно пересчитать удобство всех вагонов поезда.

Гарантируется, что в любой момент времени времени длина поезда не превышает 10^9 , а все числа A_i не превышают 10^{18} .

Формат выходных данных

После каждого из m запросов выведите два целых числа: j и A_j — номер наиболее близкого к голове поезда вагона, что его значение A_j минимально, и само значение A_j .

Пример

ввод	вывод
1 8	1 0
1 1	1 1
3 1 1	1 2
3 1 1	3 0
2 1	3 0
2 1	1 3
3 1 1	5 0
2 1	1 4
3 1 5	

Пояснение

- Изначально поезд состоит из одного вагона $A_1 = 0$, обозначим для простоты его как $[0]$.
- После прицепления одного вагона к голове, получается поезд $[0, 0]$.
- После уточнения с параметрами $b = 1, s = 1$, получается поезд $[1, 2]$.
- После ещё одного уточнения с параметрами $b = 1, s = 1$, получается поезд $[2, 4]$.
- После подцепления одного вагона в конец, получается поезд $[2, 4, 0]$.
- После ещё одного подцепления одного вагона в конец, получается поезд $[2, 4, 0, 0]$.
- После уточнения с параметрами $b = 1, s = 1$, получается поезд $[3, 6, 3, 4]$.
- После подцепления одного вагона в конец, получается поезд $[3, 6, 3, 4, 0]$.
- После уточнения с параметрами $b = 1, s = 5$, получается поезд $[4, 12, 14, 20, 21]$.

Система оценки

Тесты к этой задаче состоят из четырёх групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Дополнительные ограничения		Комментарий
		n	m	
0	0	–	–	Тесты из условия
1	27	$1 \leq n \leq 5000$	$1 \leq m \leq 5000$	Сумма k по всем запросам не превосходит 5000
2	27	–	$1 \leq m \leq 5000$	
3	19	–	–	Запросов 1 и 2 типов суммарно не более 5000
4	27	–	–	Offline-проверка

Задача Dezhnev. Путешествие по музеям

Имя входного файла:	<code>input.txt</code> или стандартный поток ввода
Имя выходного файла:	<code>output.txt</code> или стандартный поток вывода
Ограничение по времени:	4 секунды
Ограничение по памяти:	512 мегабайт

В стране N есть n городов, соединённых m односторонними дорогами. Эта, казалось бы, непримечательная страна всё же интересна двумя своими особенностями. Во-первых, неделя здесь длится d дней. Во-вторых, в каждом городе страны N расположен ровно один музей.

Турфирма «Открытые музеи» разрабатывает новую программу для путешественников, интересующихся музеями. Работникам турфирмы известно, в какие дни недели каждый из музеев открыт для посещения. Путешествие должно начинаться в столице — городе с номером 1, причём день начала путешествия должен быть первым днём недели. Днём турист будет находиться в городе и смотреть экспозицию музея (в случае если музей, конечно, работает сегодня), а в конце дня путешествие либо заканчивается, либо турист отправляется в другой город, соединённый с текущим дорогой. Дорожная система N устроена так, что перемещение по одной дороге всегда занимает одну ночь, кроме того все дороги в стране **односторонние**. Во время путешествия разрешается посещать один город несколько раз.

Вам требуется разработать такой маршрут для путешествия, что количество **различных** музеев, которые можно посетить за его время, было бы как можно больше.

Формат входных данных

Первая строка содержит три целых числа n , m и d ($1 \leq n \leq 100\,000$, $0 \leq m \leq 100\,000$, $1 \leq d \leq 50$) — количество городов, количество дорог и количество дней в неделе.

Каждая из следующих m строк содержит два целых числа u_i и v_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$), обозначающие **одностороннюю** дорогу из города u_i в город v_i .

Следующие n строк содержат график работы музеев. График работы музея, находящегося в i -м городе, описывается в i -й из этих строк. Каждая строка состоит ровно из d символов «0» или «1», j -й символ строки равен «1», если музей открыт в j -й день недели, и «0» в противном случае.

Гарантируется, для каждой пары городов (u, v) существует не более одной дороги, ведущей из u в v .

Формат выходных данных

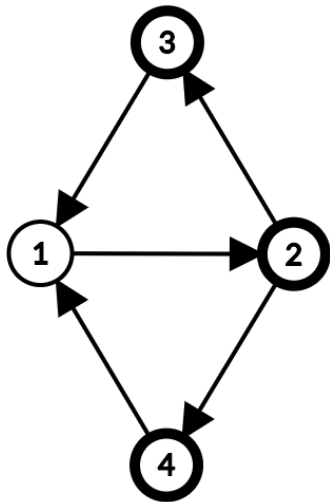
Выведите одно целое число — максимальное количество различных музеев, которые можно посетить, начав путешествие в первом городе в первый день недели.

Примеры

ВВОД	ВЫВОД
4 5 3 3 1 1 2 2 4 4 1 2 3 011 110 111 001	3
3 3 7 1 2 1 3 2 3 1111111 0000000 0111111	2

Пояснение

Пояснение к первому примеру



Максимально возможное количество музеев для посещения равно 3. Можно посетить 3 различных музея, например, так, как описано ниже.

День 1. Сейчас день недели 1, и турист находится в городе 1. Музей там закрыт. Ночью турист направляется в город 2.

День 2. Сейчас день недели 2, и турист находится в городе 2. Музей там открыт, турист его посещает. Ночью турист направляется в город 4.

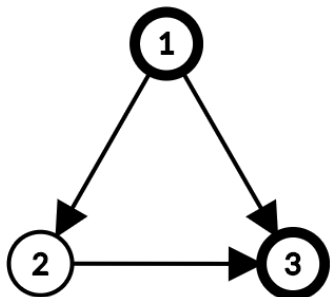
День 3. Сейчас день недели 3, и турист находится в городе 4. Музей там открыт, турист его посещает. Ночью турист направляется в город 1.

День 4. Сейчас день недели 1, и турист находится в городе 1. Музей там закрыт. Ночью турист направляется в город 2.

День 5. Сейчас день недели 2, и турист находится в городе 2. Музей там открыт, но турист в нём уже был. Ночью турист направляется в город 3.

День 6. Сейчас день недели 3, и турист находится в городе 3. Музей там открыт, турист его посещает. На этом путешествие заканчивается.

Пояснение ко второму примеру



Максимально возможное количество музеев для посещения равно 2. Можно посетить 2 различных музея, например, так, как описано ниже.

День 1. Сейчас день недели 1, и турист находится в городе 1. Музей там открыт, турист его посещает. Ночью направляемся в город 2.

День 2. Сейчас день недели 2, и турист находится в городе 2. Музей там закрыт. Ночью направляемся в город 3.

День 3. Сейчас день недели 3, и турист находится в городе 3. Музей там открыт, турист его посещает. На этом путешествие заканчивается.

Система оценки

Тесты к этой задаче состоят из шести групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **необходимых** групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Открытая олимпиада школьников по программированию 2018/19, второй тур
Москва, Сочи, 8 марта 2019

Группа	Баллы	Дополнительные ограничения			Необх. группы	Комментарий
		n	d	График работы		
0	0	–	–	–	–	Тесты из условия.
1	9	$n \leq 10$	$d \leq 10$	–	0	
2	12	$n \leq 100$	$d \leq 10$	–	0, 1	
3	13	–	–	Каждый музей работает ровно один день в неделе	–	
4	17	–	–	Каждый музей не работает ровно один день в неделе	–	
5	13	–	$d = 2$	–	–	
6	36	–	–	–	0, 1, 2, 3, 4, 5	Offline-проверка

Задача Gagarin. Расписание смены

Имя входного файла:	input.txt или стандартный поток ввода
Имя выходного файла:	output.txt или стандартный поток вывода
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

В известной на всю страну Весенней компьютерной школе скоро начнется новая смена, и в связи с этим весь дружный состав преподавателей и кураторов начал составлять ее расписание. В процессе обсуждения они пришли к расписанию s , которое может быть представлено в виде бинарной строки, в которой на i -й позиции стоит «1», если ученики в i -й день пишут констест, и «0», если отдыхают.

Расписание уже почти утвердили, но в этот момент на заседание пришел главный методист Глеб и заявил, что за весь свой большой преподавательский опыт он заметил одну особенность — если смена в школе проходит по расписанию t (которое может быть описано в таком же формате, что и s), то ученики особенно хорошо усваивают материал. Поскольку количество дней в грядущей смене может отличаться от количества дней в t , Глеб потребовал, чтобы уже составленное расписание переделали таким образом, чтобы количество вхождений t в него как подстроки было максимально — он считает, что таким образом эффективность обучения в школе будет наибольшей. При этом количество учебных и выходных дней не должно измениться, может измениться только порядок их следования.

Поскольку у преподавателей, кураторов и методистов и так много дел (первым нужно составить констесты, вторым придумать развлекательную программу, а последние заняты думами о методике), они поручили это дело вам. Сможете ли вы переделать расписание оптимальным образом?

Формат входных данных

Первая строка содержит строку s ($1 \leq |s| \leq 500\,000$), задающую текущий проект расписания смены.

Вторая строка содержит строку t ($1 \leq |t| \leq 500\,000$), задающую оптимальное расписание согласно Глебу.

Строки s и t состоят только из символов «0» и «1».

Формат выходных данных

В единственной строке выведите расписание, количество вхождений t как подстроки в которое максимально. Выведенное расписание должно состоять только из «0» и «1», причём количество нулей должно совпадать с количеством нулей в s , а количество единиц — с количеством единиц в s .

Если существует несколько оптимальных расписаний, выведите любое из них.

Примеры

ввод	вывод
101101 110	110110
10010110 100011	01100011
10 11100	01

Пояснение

В первом примере два вхождения, начинающихся с первой и четвертой позиции.

Во втором примере всего одно вхождение, и оно начинается с третьей позиции. Заметим также, что ответ не единственен — например, самый первый день (являющийся выходным) можно переместить на последнюю позицию, и кол-во вхождений t не изменится.

В третьем примере добиться даже одного вхождения не получится.

Система оценки

Тесты к этой задаче состоят из четырёх групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Дополнительные ограничения	Комментарий
		$ s , t $	
0	0	–	Тесты из условия.
1	16	$ s , t \leq 8$	
2	27	$ s , t \leq 50$	
3	36	$ s , t \leq 5000$	
4	21	–	Offline-проверка