

Задача А. Серебряный статус в авиакомпании

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

В авиакомпании Б для того, чтобы достичь в бонусной программе серебряного статуса, необходимо за текущий календарный год накопить 20 000 квалификационных миль, которые начисляют за совершённые перелёты. Мили прошлых лет, а также мили, полученные за покупки у партнёров, квалификационными не являются. Если квалификационных миль для присвоения статуса не хватает, то в конце года любые из имеющихся миль можно обменять на квалификационные. А именно, 7 500 миль можно обменять на 1 000 квалификационных, а 30 000 миль можно обменять на 5 000 квалификационных.

Евгений Васильевич накопил n миль, k из которых являются квалификационными. Если количества достаточно, чтобы с учётом возможных обменов достичь серебряного статуса, то менять свои мили на квалификационные Евгений Васильевич хочет так, чтобы в результате у него осталось как можно больше миль. Помогите ему — определите, какое максимальное общее количество миль будет у Евгения Васильевича после обменов, необходимых для получения как минимум 20 000 квалификационных миль, или определите, что с имеющимися милями достичь серебряного статуса невозможно.

Формат входных данных

Первая строка содержит одно целое число n ($0 < n \leq 200\,000$) — общее число миль, накопленных Евгением Васильевичем.

Вторая строка содержит одно целое число k ($0 \leq k \leq n$) — количество квалификационных миль среди накопленных n миль.

Формат выходных данных

Выведите одно целое число — максимальное количество миль, которое будет у Евгения Васильевича после оптимальных обменов, необходимых для получения как минимум 20 000 квалификационных миль.

В случае, если серебряного статуса достичь невозможно, выведите -1 .

Система оценки

В данной задаче 20 тестов, помимо тестов из условия, каждый из них оценивается в 5 баллов. Результаты работы ваших решений на первых 20 тестах будут доступны во время соревнования. Результаты работы на остальных 5 тестах будут доступны после окончания соревнования.

Решения, корректно работающие **только** в случаях, когда серебряного статуса достичь невозможно, оцениваются в 0 баллов.

Примеры

стандартный ввод	стандартный вывод
35000 18000	22000
50000 16000	25000
26500 19001	20000
21000 1000	-1

Замечание

В первом примере до серебряного статуса не хватает 2 000 квалификационных миль, для их

получения нужно дважды обменять 7 500 миль на 1 000 квалификационных. То есть будут потрачены 15 000 миль, и 2 000 миль будут приобретены.

Во втором примере выгоднее обменять 30 000 миль на 5 000 квалификационных.

В третьем примере для обмена придётся потратить и одну квалификационную милю.

В четвертом примере достичь серебряного статуса нельзя.

Задача В. Разрез таблицы

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

Участникам, использующим язык Python3, рекомендуется отправлять решения на проверку с использованием интерпретатора PyPy3.

Дана таблица, состоящая из n строк и m столбцов. В каждой ячейке таблицы записан 0 или 1. Необходимо разделить таблицу на две части разрезом, проходящим из левого верхнего угла границы таблицы в правый нижний. Линия разреза может идти только вдоль границ ячеек таблицы в направлении вправо или вниз (см. рисунки ниже).

Пусть a — количество единиц в одной части таблицы после разреза, b — количество единиц в другой части таблицы. Требуется максимизировать величину $a \cdot b$.

Формат входных данных

В первой строке содержится два целых числа n и m ($1 \leq n, m \leq 3 \cdot 10^5$, $2 \leq n \cdot m \leq 3 \cdot 10^5$) — количество строк и столбцов в таблице соответственно.

В каждой из следующих n строк записаны m целых чисел, j -е число в i -й из этих строк соответствует значению $a_{i,j}$ ($0 \leq a_{i,j} \leq 1$).

Формат выходных данных

В первой строке выведите единственное число — максимальное значение произведения.

Если вы не хотите предъявлять разрез, при котором достигается ответ, то выведите во второй строке символ '-'. Иначе, во второй строке выведите строку, состоящую из n символов 'D' и m символов 'R', обозначающие очередное направление линии разреза длиной один, где 'D' означает разрез вниз, а 'R' — разрез вправо.

Обратите внимание, что ответ может быть больше, чем возможное значение 32-битной целочисленной переменной, поэтому необходимо использовать 64-битные целочисленные типы данных (тип `int64` в языке Pascal, тип `long long` в C и C++, тип `long` в Java и C#). Язык Python будет корректно работать и с типом `int`.

Система оценки

В данной задаче 25 тестов, каждый из которых оценивается в 4 балла. Результаты работы ваших решений на первых 20 тестах будут доступны во время соревнования. Результаты работы на остальных 5 тестах будут доступны после окончания соревнования.

Вы получите 1 балл за тест, если верно определите максимальное значение произведения. Оставшиеся 3 балла вы получите в случае верно восстановленного разреза.

Гарантируется, что решения корректно работающие при $1 \leq n, m \leq 7$, наберут не менее 16 баллов.

Гарантируется, что решения корректно работающие при $n \leq 2$ или $m \leq 2$, наберут не менее 20 баллов.

Гарантируется, что решения корректно работающие при $2 \leq n \cdot m \leq 700$, наберут не менее 32 баллов.

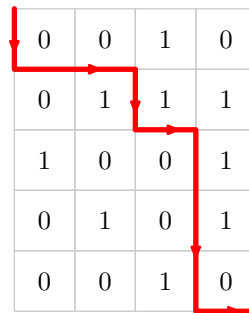
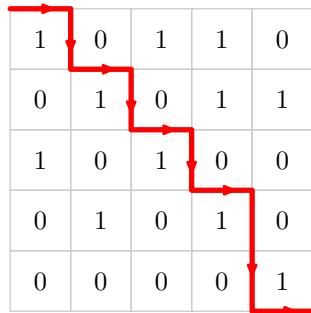
Гарантируется, что решения корректно работающие в случаях, когда каждая строка и столбец имеет не больше одного ненулевого значения, наберут не менее 16 баллов.

Примеры

стандартный ввод	стандартный вывод
5 5 1 0 1 1 0 0 1 0 1 1 1 0 1 0 0 0 1 0 1 0 0 0 0 0 1	30 RDRDRDRDDR
5 4 0 0 1 0 0 1 1 1 1 0 0 1 0 1 0 1 0 0 1 0	20 DRRDRDDDR

Замечание

На рисунках изображены корректные разрезы для каждого примера, при которых достигается максимальное значение произведения.



Задача С. Любовь к комбинаторным объектам

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

Саша очень любит комбинаторику и считает, что в любом контексте должна быть задача на неё. К несчастью, ни его друзья, ни друзья его друзей так не считают.

После того, как Саша дал задачу на комбинаторику на Очень Важную Олимпиаду, всеобщее терпение рухнуло, и теперь каждый хочет публично озвучить свою неприязнь к комбинаторике. Саша, с другой стороны, готов её защищать, и если кто-то из его друзей напишет ему, что задач на комбинаторику не должно быть на школьных олимпиадах, Саша проведёт с ним воспитательную беседу, и начиная со следующего дня этот человек сам начнёт защищать комбинаторику и проводить воспитательные беседы со своими друзьями.

Всего есть n человек, пронумерованных от 1 до n , Саша — имеет номер 1. Некоторые пары людей дружат друг с другом. Человек под номером i ($2 \leq i \leq n$) будет писать всем своим друзьям о своей нелюбви к комбинаторике в дни $a_i, a_i + t_i, a_i + 2t_i$ и так далее, пока не окажется, что кто-то из его друзей в этот день уже защищает комбинаторику. В этом случае с ним **в тот же день** проведут воспитательную беседу, и, **начиная со следующего дня**, он будет защищать комбинаторику. Саша защищает комбинаторику начиная со дня 1.

Для каждого человека, кроме Саши, определите, в какой день с ним проведут воспитательную беседу.

Формат входных данных

Первая строка содержит два целых числа n и m ($2 \leq n \leq 100\,000, 1 \leq m \leq 100\,000$) — количество людей, включая Сашу, и количество дружеских связей между людьми.

Вторая строка содержит $n - 1$ целых чисел a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — моменты времени, когда каждый из людей начинает писать своим друзьям.

Третья строка содержит $n - 1$ целых чисел t_2, \dots, t_n ($1 \leq t_i \leq 10^9$) — периоды, с которыми каждый человек пишет своим друзьям.

В каждой из следующих m строк содержится два целых числа u_i и v_i ($1 \leq u_i, v_i \leq n$) — номера дружащих людей.

Гарантируется, что пары дружащих людей не повторяются, то есть $(u_i, v_i) \neq (u_j, v_j)$ и $(u_i, v_i) \neq (v_j, u_j)$, если $i \neq j$.

Гарантируется, что в результате процесса, описанного в условии, с каждым человеком в конце концов проведут воспитательную беседу.

Формат выходных данных

В единственной строке выведите $n - 1$ чисел — номера дней, когда с человеком под номером i ($2 \leq i \leq n$), проведут воспитательную беседу.

Система оценки

Тесты к этой задаче состоят из девяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп. Обратите внимание, что прохождение тестов из условия не требуется для некоторых групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования. Группы в данной задаче «склеиваются». Это означает, что за подгруппу начисляются баллы, если хотя бы одна из ваших посылок успешно проходит все её тесты. Обратите внимание, что если у подгруппы имеются необходимые подгруппы, то для проверки на данной подгруппе текущая посылка должна пройти тесты во всех этих необходимых подгруппах (при этом результаты предыдущих посылок не учитываются).

Группа	Баллы	Доп. ограничения		Необх. группы	Комментарий
		n, m	a_i, t		
0	0	–	–	–	Тесты из условия
1	6	$n \leq 10$	–	0	–
2	7	$m = n - 1$	–	–	–
3	8	–	$a_i = t_i = 1$	–	–
4	5	–	$t_i = 1$	3	–
5	11	–	$a_i, t_i \leq 10$	0, 3	–
6	14	$n, m \leq 300$	$a_i, t_i \leq 300$	0	–
7	13	$n, m \leq 4000$	$a_i, t_i \leq 4000$	0, 6	–
8	15	$n, m \leq 4000$	–	0, 6, 7	–
9	21	–	–	0–8	Offline-проверка

Примеры

стандартный ввод	стандартный вывод
3 2 3 2 3 2 1 2 3 2	3 4
5 5 7 1 3 2 6 1 2 4 1 2 2 3 3 4 4 1 3 5	7 4 3 6

Замечание

В первом тестовом примере:

- Человек 1 — это Саша.
- Человек 2 собирается писать своим друзьям, Саше и человеку 3, в дни 3, 6, 9, ...
- Человек 3 собирается писать своему другу, человеку 2, в дни 2, 4, 6, ...

Процесс выглядит так:

- В день 2 человек 3 пишет человеку 2.
 - Человек 2 ещё не защищает комбинаторику, поэтому ничего не происходит.
- В день 3 человек 2 пишет Саше и человеку 3.
 - Саша проводит со человеком 2 воспитательную беседу.
- В день 4 человек 3 снова пишет человеку 2.
 - Начиная со этого дня человек 2 защищает комбинаторику, поэтому он проводит с человеком 3 воспитательную беседу.

Задача D. Черепашка наносит ответный удар

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Участникам, использующим язык Python3, рекомендуется отправлять решения на проверку с использованием интерпретатора PyPy3.

После тяжёлой тренировки Микеланджело и Рафаэль решили заказать пиццу. Сегодня, по случаю праздника, пиццерия готовит прямоугольные пиццы из n рядов и m колонок, разрезанные на $n \cdot m$ квадратных кусочков. Каждый кусочек имеет свой уникальный рецепт и вкус.

Микеланджело первым открыл коробку и оценил удовольствие от каждого кусочка: удовольствие от куска, находящегося в i -м ряду и j -м столбце, равно $a_{i,j}$. Понравившиеся куски он оценивает неотрицательными числами, не понравившиеся — отрицательными. Гарантируется, что есть хотя бы один кусочек, который понравился Микеланджело, то есть среди $a_{i,j}$ есть хотя бы одно неотрицательное число.

Черепашки договорились, что первым есть будет Микеланджело. По старинной традиции он должен выбрать маршрут от верхнего левого угла $(1, 1)$ до нижнего правого угла (n, m) и съесть **все** кусочки на этом маршруте. За один шаг он может перейти либо в соседний кусок справа, либо в соседний кусок снизу.

Микеланджело стремится максимизировать суммарное удовольствие от съеденных кусочков.

Однако Рафаэль решил воспользоваться соусом. **До того** как Микеланджело выберет маршрут, Рафаэль решил выбрать **ровно один** кусочек пиццы с неотрицательным значением и нанести на него соус. К сожалению, Микеланджело не любит этот специфический соус, поэтому удовольствие от этого кусочка меняется на противоположное: если раньше оно было равно $a_{i,j}$, то после соуса становится $-a_{i,j}$.

После этого Микеланджело, зная выбор Рафаэля, выберет оптимальный для себя маршрут и съест все куски на нём.

Рафаэлю стало интересно, какое минимальное удовольствие Микеланджело может получить в зависимости от выбора Рафаэля. Помогите ему, посчитайте это число за него.

Формат входных данных

В первой строке заданы два целых числа n, m ($1 \leq n, m \leq 10^6$, $1 \leq n \cdot m \leq 10^6$) — количество строк и столбцов в пицце соответственно.

В каждой из следующих n строк записаны m целых чисел, j -е число в i -й из этих строк соответствует значению $a_{i,j}$ ($-10^9 \leq a_{i,j} \leq 10^9$).

Формат выходных данных

Выведите единственное целое число — минимальное удовольствие Микеланджело, которое Рафаэль может гарантировать.

Обратите внимание, что ответ может быть больше, чем возможное значение 32-битной целочисленной переменной, поэтому необходимо использовать 64-битные целочисленные типы данных (тип `int64` в языке Pascal, тип `long long` в C и C++, тип `long` в Java и C#). Язык Python будет корректно работать и с типом `int`.

Система оценки

В данной задаче 50 тестов, помимо тестов из условия, каждый из них оценивается в 2 балла. Результаты работы ваших решений на всех тестах будут доступны сразу во время соревнования.

Решения, корректно работающие при $n \cdot m \leq 5000$, наберут не менее 22 баллов.

Решения, корректно работающие при $n, m \leq 500$, будут наберут не менее 26 баллов.

Пример

стандартный ввод	стандартный вывод
3 3 1 -2 3 4 -5 2 1 6 -1	3

Замечание

Если бы Рафаэль не использовал соус, Микеланджело выбрал бы путь

$$(1, 1) \rightarrow (2, 1) \rightarrow (3, 1) \rightarrow (3, 2) \rightarrow (3, 3),$$

который даёт суммарное удовольствие $1 + 4 + 1 + 6 + (-1) = 11$.

Однако, если Рафаэль нанесёт соус на клетку $(3, 2)$, значение 6 превратится в -6 . Тогда оптимальным для Микеланджело станет путь

$$(1, 1) \rightarrow (1, 2) \rightarrow (1, 3) \rightarrow (2, 3) \rightarrow (3, 3),$$

с суммарным удовольствием $1 + (-2) + 3 + 2 + (-1) = 3$. Ни в какой другой клетке Рафаэль не может добиться меньшего результата для Микеланджело.

Задача Е. Составляем многоугольник

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Антон всё еще не любит легенды, зато любит геометрию!

В ряд лежат n отрезков, длина i -го отрезка равна a_i . Вам нужно посчитать количество пар (l, r) , таких что:

- $1 \leq l \leq r \leq n$
- Можно составить невырожденный многоугольник, в котором длины сторон являются некоторой перестановкой подотрезка a_l, a_{l+1}, \dots, a_r .

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число t ($1 \leq t \leq 500$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит одно целое число n ($1 \leq n \leq 2 \cdot 10^5$) — количество отрезков.

Вторая строка каждого набора входных данных содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — длину каждого из отрезков.

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

Формат выходных данных

Для каждого набора входных данных выведите одно целое число — искомое количество пар (l, r) .

Система оценки

Тесты к этой задаче состоят из пяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп. Обратите внимание, что прохождение тестов из условия не требуется для некоторых групп. Группы в данной задаче «склеиваются». Это означает, что за подгруппу начисляются баллы, если хотя бы одна из ваших посылок успешно проходит все её тесты. Обратите внимание, что если у подгруппы имеются необходимые подгруппы, то для проверки на данной подгруппе текущая посылка должна пройти тесты во всех этих необходимых подгруппах (при этом результаты предыдущих посылок не учитываются).

Обозначим за N сумму n по всем тестовым случаям.

Группа	Баллы	Доп. ограничения	Необх. группы	Комментарий
0	0	–	–	Тесты из условия
1	16	$N \leq 200$	0	–
2	12	$N \leq 5000$	0, 1	–
3	22	$a_i \leq a_{i+1}$ для всех $1 \leq i \leq n - 1$	–	–
4	27	Гарантируется, что при генерации массива a жюри сначала сгенерировало массив, а затем случайно перемешало его элементы	–	–
5	23	–	0–5	–

Пример

стандартный ввод	стандартный вывод
4	6
5	9
1 1 1 1 1	14
7	58
2 1 3 2 9 5 7	
7	
1 2 3 4 5 6 7	
15	
1 2 1 4 1 2 1 8 1 2 1 4 1 2 1	

Замечание

В первом наборе входных данных подойдут следующие пары: (1, 3), (1, 4), (1, 5), (2, 4), (2, 5), (3, 5).

Во втором наборе входных данных подойдут следующие пары: (1, 4), (1, 6), (1, 7), (2, 6), (2, 7), (3, 6), (3, 7), (4, 7), (5, 7).

Задача F. Опять деревья...

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	1024 мегабайта

Дано дерево, состоящее из n вершин. На каждой вершине записано целое неотрицательное число a_v . Также вам даны k различных целых неотрицательных чисел b_1, \dots, b_k .

Иван изучает рёбра данного дерева. Он считает множество рёбер красивым, если после удаления этих рёбер из дерева дерево распадается на компоненты связности, где в каждой из компонент побитовый XOR всех чисел a_v в этой компоненте принадлежит множеству b .

Напомним, что операция побитового XOR (\oplus) обозначается «xor» в Паскале, «^» в C++, Python и Java; для двух целых чисел она определена следующим образом: i -й бит результата равен 1 тогда и только тогда, когда в одном из чисел этот бит 1, а в другом 0. Например, $6 \oplus 10 = 0110_2 \oplus 1010_2 = 1100_2 = 12$.

Посчитайте количество множеств рёбер в дереве, которые Иван считает красивыми. Поскольку это число может быть слишком большим, от вас требуется вывести только остаток от деления этого числа на $10^9 + 7$.

Формат входных данных

Первая строка содержит два целых числа n и k ($2 \leq n \leq 10^5$, $1 \leq k \leq 10$) — количество вершин в дереве и размер множества b .

Следующие $n - 1$ строк описывают рёбра дерева. i -я из них содержит два целых числа v_i и u_i ($1 \leq v_i, u_i \leq n$) — номера вершин, соединённых i -м рёбером.

Следующая строка содержит n целых чисел a_1, a_2, \dots, a_n ($0 \leq a_i < 2^{30}$) — значения, записанные на вершинах.

Следующая строка содержит k целых различных чисел b_1, b_2, \dots, b_k ($0 \leq b_i < 2^{30}$) — элементы множества b .

Формат выходных данных

В единственной строке выведите ответ на задачу.

Система оценки

Тесты к этой задаче состоят из девяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп. Обратите внимание, что прохождение тестов из условия не требуется для некоторых групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования. Группы в данной задаче «склеиваются». Это означает, что за подгруппу начисляются баллы, если хотя бы одна из ваших посылок успешно проходит все её тесты. Обратите внимание, что если у подгруппы имеются необходимые подгруппы, то для проверки на данной подгруппе текущая посылка должна пройти тесты во всех этих необходимых подгруппах (при этом результаты предыдущих посылок не учитываются).

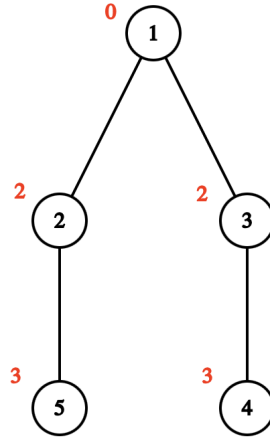
Группа	Баллы	Доп. ограничения		Необх. группы	Комментарий
		n	k		
0	0	–	–	–	Тесты из условия
1	13	–	$k = 1$	–	$b_1 = 0$
2	11	$n \leq 15$	$k \leq 4$	0	–
3	15	–	$k \leq 4$	0	$a_i, b_i < 32$
4	15	–	$k \leq 4$	–	$v_i = i, u_i = i + 1$
5	9	–	$k = 1$	1	–
6	11	–	$k \leq 2$	0, 1, 5	–
7	12	–	$k \leq 4$	0–6	–
8	8	–	$k \leq 6$	0–7	–
9	4	–	$k \leq 8$	0–8	Offline-проверка
10	2	–	–	0–9	Offline-проверка

Примеры

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

Замечание

Иллюстрация к первому примеру. В нем можно удалить ребро между вершинами 1 и 2. Тогда побитовый XOR в компоненте, состоящей из вершин 2 и 5 равен $a_2 \oplus a_5 = 2 \oplus 3 = 1$. Побитовый XOR в компоненте вершин 1, 3 и 4 равен $a_1 \oplus a_3 \oplus a_4 = 0 \oplus 2 \oplus 3 = 1$. Вторым красивым набором является множество, состоящее из ребра, соединяющего вершины 1 и 3.



В третьем примере красивыми будут множества (ребро обозначается парой вершин, которые оно соединяет) $[(1, 2)], [(1, 3)], [(2, 5)], [(3, 4)]$. Обратите внимание, что первый и третий примеры отличаются только множеством b .

В четвертом примере красивыми будут множества $[(1, 2)], [(3, 4)], [(1, 2), (2, 3), (3, 4)]$.