

Задача А. Вечная Копилка

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

У вас есть n гаджетов, которые вы хотите использовать одновременно.

Гаджет номер i использует a_i единиц энергии в секунду. Это использование энергии непрерывное, то есть, за λ секунд гаджет использует $\lambda \cdot a_i$ единиц энергии. Гаджет номер i сейчас имеет b_i единиц энергии. Все устройства могут хранить неограниченное количество единиц энергии.

У вас есть единственное зарядное устройство, которое можно подключить к любому из гаджетов. Зарядное устройство заряжает подключенный гаджет на p единиц энергии в секунду. Зарядка происходит непрерывно, то есть, за λ секунд зарядки любой гаджет получит $\lambda \cdot p$ единиц энергии. Вы можете подключать зарядное устройство к любому гаджету в любой момент времени (в том числе, в нецелые моменты времени), можно считать, что переключение происходит мгновенно.

Вы задались вопросом: как долго вы можете использовать все гаджеты, прежде чем в каком-то из них останется 0 единиц энергии?

Если вы можете использовать гаджеты бесконечно долго, выведите -1 . Иначе выведите максимально возможное время, в течение которого во всех гаджетах будет больше 0 единиц энергии.

Формат входных данных

Первая строка содержит два целых числа n и p ($1 \leq n \leq 100000, 1 \leq p \leq 10^9$) — количество гаджетов и мощность зарядного устройства.

После этого следуют n строк, каждая из которых содержит два целых числа. Строка номер i содержит числа a_i и b_i ($1 \leq a_i, b_i \leq 100000$) — мощность устройства и запас энергии в устройстве в начальный момент времени.

Формат выходных данных

Если вы можете использовать гаджеты бесконечно долго, выведите -1 . Иначе выведите максимально возможное время, в течение которого во всех гаджетах будет больше 0 единиц энергии.

Ваш ответ будет считаться правильным, если его абсолютная или относительная ошибка не превосходит 10^{-4}

Примеры

стандартный ввод	стандартный вывод
2 1 2 2 2 1000	2.0000000000
1 100 1 1	-1
3 5 4 3 5 2 6 1	0.5000000000

Задача В. Слишком простые задачи

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

Вы готовитесь к экзамену по теории расписаний. Экзамен продлится ровно T миллисекунд. На экзамене вас ждут n задач. Каждую задачу номер i вы сможете либо полностью решить, затратив на это t_i миллисекунд, либо не решить вовсе, не потратив на нее ни миллисекунды. Времени на отдых после решения задачи вам не потребуется.

К несчастью, ваш преподаватель считает, что некоторые задачи слишком просты для вас, поэтому каждой задаче i сопоставил целое число a_i , обозначающее, что задача i может принести балл за ее решение только в том случае, если вы решите не более a_i задач всего (включая задачу i).

Формально, пусть за экзамен вы решите некоторое множество задач p_1, p_2, \dots, p_k . Ваш итоговый балл за экзамен s будет равен количеству таких j от 1 до k , что $k \leq a_{p_j}$.

Догадавшись, что настоящая первая задача экзамена уже перед вами, вы хотите заранее выбрать подмножество задач для решения на экзамене, которое принесет вам наибольший возможный итоговый балл. Не забудьте, что экзамен ограничен по времени, и вам должно хватить времени на решение всех выбранных задач. Если существуют разные множества задач, ведущие к наибольшему итоговому баллу, вас устроит любое из них.

Формат входных данных

Первая строка содержит два целых числа n и T ($1 \leq n \leq 2 \cdot 10^5$; $1 \leq T \leq 10^9$) — количество задач на экзамене и длительность экзамена в миллисекундах, соответственно.

Каждая из следующих n строк содержит пару целых чисел a_i и t_i ($1 \leq a_i \leq n$; $1 \leq t_i \leq 10^4$). Задачи пронумерованы от 1 до n .

Формат выходных данных

В первой строке выведите целое число s — наибольший возможный итоговый балл.

Во второй строке выведите целое число k ($0 \leq k \leq n$) — количество задач, которые вам следует решить.

В третьей строке выведите k различных целых чисел p_1, p_2, \dots, p_k ($1 \leq p_i \leq n$) — номера задач, которые вам следует решить, в любом порядке.

Если наилучших множеств задач несколько, выведите любое из них.

Примеры

стандартный ввод	стандартный вывод
5 300	2
3 100	2
4 150	3 4
4 80	
2 90	
2 300	
2 100	0
1 787	0
2 788	
2 100	2
2 42	2
2 58	1 2

Задача С. Гордыня

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

Вам дан массив a длиной n , вы можете выполнять определенные операции над ним. Каждая операция выглядит следующим образом: выберите два соседних элемента из a , пусть это будут x и y , и замените один из них величиной $\gcd(x, y)$, где \gcd обозначает наибольший общий делитель.

Какое минимальное число операций необходимо, чтобы сделать все элементы массива равными 1?

Формат входных данных

Первая строка содержит одно целое число n ($1 \leq n \leq 2000$) — количество элементов в массиве.

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — элементы массива.

Формат выходных данных

Выведите -1, если невозможно сделать все элементы массива равными 1. Иначе выведите минимальное число операций, необходимых для того, чтобы сделать все числа равными 1.

Примеры

стандартный ввод	стандартный вывод
5 2 2 3 4 6	5
4 2 4 6 8	-1
3 2 6 9	4

Задача D. Идеальные группы

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

SaMeг написал лучший тест всех времён и народов для одной из его задач. В этой задаче требуется для данного массива целых чисел найти минимальное число групп, на которое можно разбить данный массив так, что произведение любых двух чисел в одной группе — точный квадрат.

Каждое целое число должно находиться ровно в одной группе, однако числа в группе не обязательно должны быть последовательными в массиве.

SaMeг хочет создать больше тестов из того теста, который у него уже есть. Его тест является массивом A из n целых чисел и ему необходимо найти число подмассивов A , состоящих из чисел, идущих в изначальном массиве последовательно, таких, что ответ на задачу для подмассива будет равен k , для всех k от 1 до n включительно.

Формат входных данных

В первой строке содержится одно целое число n ($1 \leq n \leq 5000$) — размер массива.

Во второй строке содержится n целых чисел a_1, a_2, \dots, a_n ($-10^8 \leq a_i \leq 10^8$) — значения чисел в массиве.

Формат выходных данных

Выведите n целых чисел, разделённых пробелами, k -е из которых должно быть равно числу подмассивов A , состоящих из чисел, идущих в изначальном массиве последовательно, ответ для которых на задачу равен k .

Примеры

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

Задача Е. Очередная задача на подпоследовательности

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

Последовательность целых чисел a_1, a_2, \dots, a_k называется хорошим массивом, если $a_1 = k - 1$ и $a_1 > 0$. Например последовательности $[3, -1, 44, 0]$, $[1, -99]$ — хорошие массивы, а последовательности $[3, 7, 8]$, $[2, 5, 4, 1]$, $[0]$ — нет.

Последовательность целых чисел называется хорошей, если она состоит из положительного количества подряд идущих хороших массивов. Например последовательности $[2, -3, 0, 1, 4]$, $[1, 2, 3, -3, -9, 4]$ — хорошие, а последовательности $[2, -3, 0, 1]$, $[1, 2, 3, -3, -9, 4, 1]$ — нет.

Для заданной последовательности чисел подсчитайте количество её подпоследовательностей, которые являются хорошими последовательностями, по модулю 998244353.

Формат входных данных

В первой строке задано число n ($1 \leq n \leq 10^3$) — длина изначальной последовательности. Следующая строка содержит n целых чисел a_1, a_2, \dots, a_n ($-10^9 \leq a_i \leq 10^9$) — сама последовательность.

Формат выходных данных

В единственной строке выведите число — количество подпоследовательностей исходной последовательности, которые являются хорошими последовательностями, по модулю 998244353.

Примеры

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

Задача F. Ральф и его магическое поле

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

У Ральфа есть магическое поле, которое разделено на $n \times m$ блоков. А именно, на поле есть n строк и m столбцов. Ральф хочет в каждый блок поставить целое число. Однако, магическое поле не всегда работает правильно. Оно работает, только если произведение чисел в каждой строке и в каждом столбце равно k , где k может быть равно 1 или -1.

Ральф хочет узнать количество способов поставить некоторые целые числа во все блоки так, чтобы магическое поле работало правильно. Два способа считаются различными, если и только если существует хотя бы один блок такой, что числа в нем в первом и втором способе различны. Вам необходимо вывести остаток от деления ответа на $10^9 + 7$.

Обратите внимание, нет ограничения на целые числа, которые можно ставить в блоки, однако, можно показать, что ответ конечный.

Формат входных данных

Единственная строка содержит три целых числа n , m и k ($1 \leq n, m \leq 10^{18}$, k равно 1 или -1).

Формат выходных данных

Выведите единственное число — остаток от деления ответа на $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
1 1 -1	1
1 3 1	1
3 3 -1	16

Задача G. Сжать строку

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Имеется строка s из n символов. Каждый символ — маленькая латинская буква. Вам нужно сжать эту строку, потратив как можно меньше денег.

Для сжатия вы должны представить строку s в виде конкатенации некоторого количества непустых строк: $s = t_1 t_2 \dots t_k$. Далее, i -ю из этих строк нужно сжать одним из следующих двух способов:

1) если $|t_i| = 1$, то есть текущая строка состоит всего из одного символа, то можно просто заплатить a монет;

2) если t_i является подстрокой строки $t_1 t_2 \dots t_{i-1}$, то можно заплатить b монет.

Итак, требуется по данной строке s определить наименьшее возможное количество монет, которого хватит для некоторого сжатия s .

Формат входных данных

В первой строке находятся три целых положительных числа, разделённых пробелами: n , a и b ($1 \leq n, a, b \leq 5000$) — длина строки, стоимость сжатия строки из одного символа и стоимость сжатия подстроки, которая встречалась ранее, соответственно.

Во второй строке находится строка s , состоящая из n строчных латинских букв.

Формат выходных данных

Выведите единственное число — наименьшее возможное количество монет, необходимое для сжатия.

Примеры

стандартный ввод	стандартный вывод
3 3 1 aba	7
4 1 1 abcd	4
4 10 1 aaaa	12