

Задача А. Плагиат кода

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

Два сотрудника одной известной компании Алиса и Боб предложили тимлиду два решения возникшего в критическом месте бага. Теперь Алиса подозревает, что сотрудник Боб просто взял ее код и добавил в него не влияющие на функциональность символы, чтобы создать впечатление более интенсивной работы.

Компания пишет на эзотерическом языке программирования, похожем на Malbolge, поэтому код каждого из сотрудников представляет из себя строку из маленьких латинских букв. Код Алисы — строка t , а код Боба — строка s .

Поскольку клавиатура Боба сломана, он может печатать ровно два символа за раз, то есть может вставлять в любое место строки два любых (не обязательно одинаковых) символа. После заявления Алисы о подозрении Боба в плагиате их начальник начал анализировать строки s и t , пытаясь понять, мог ли Боб получить строку s из строки t со своей сломанной клавиатурой. Для этого он пытается постепенно удалять из строки s по два соседних символа, пока не получит в итоге строку t .

Помогите выяснить, виноват ли Боб в плагиате: определите, можно ли получить строку t из строки s , вырезая из нее произвольное количество раз по два стоящих рядом символа.

Формат входных данных

В первой строке дана строка s , состоящая из маленьких латинских букв от 'a' до 'z' ($1 \leq |s| \leq 2 \cdot 10^5$).

Во второй строке дана строка t , также состоящая из маленьких латинских букв ($1 \leq |t| \leq |s|$).

Формат выходных данных

В качестве ответа выведите «YES» (без кавычек), если из s можно получить t удалениями двух символов подряд, и «NO» в противном случае.

Вы можете выводить ответ в любом регистре (например, строки «yEs», «yes», «Yes» и «YES» будут распознаны как положительный ответ).

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	20	$ s \leq 10$		полная
2	23	t состоит только из 'a'		первая ошибка
3	27	$ s \leq 1000$; $ t = s - 2$		первая ошибка
4	30	нет	1 – 3	первая ошибка

Примеры

стандартный ввод	стандартный вывод
sobaka baka	YES
sobabka baka	NO
abacaba aca	YES

Задача В. SpamGPT-4

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

Для тестирования отказоустойчивости двух лучших спам-ботов компании «LinkedOut» было решено настроить их на взаимодействие друг с другом и посмотреть, как долго они проработают в таком режиме без ошибок.

После старта оба бота отправляют друг другу по одному сообщению, после чего первый бот отправляет новое сообщение каждые a секунд, а второй — каждые b секунд. Иными словами, первый бот отправляет новое сообщение на секундах $0, a, 2a$, и так далее, а второй — на секундах $0, b, 2b$, и так далее.

Помимо этого, оба бота отправляют ответ на каждое полученное сообщение ровно спустя секунду после получения. Сообщения отправляются без задержки и приходят моментально после отправки. В частности, если в момент времени t первый бот отправит сообщение, то в момент времени $t + 1$ он получит ответ на него, а в момент времени $t + 2$ — отправит свой ответ. Также боты отлично выполняют параллельные задачи параллельно и могут отправлять любое количество сообщений одновременно (например, если надо одновременно отправить новое сообщение и ответы на полученные).

Вам даны параметры ботов a и b . Определите, сколько сообщений каждый из ботов должен будет отправить к моменту времени T , если они оба будут работать без ошибок.

Формат входных данных

В единственной строке ввода через пробел даны три целых числа a, b и T — периодичности отправки новых сообщений и время работы ботов ($1 \leq a, b, T \leq 10^9$).

Формат выходных данных

Выведите через пробел два целых числа — количество сообщений, отправленных к моменту T первым и вторым ботом, соответственно. Если какие-то сообщения должны быть отправлены в T -ю секунду, их тоже следует учесть в ответе.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	15	$a, b, T \leq 1000$		полная
2	18	$T \leq 5 \cdot 10^4$	1	первая ошибка
3	21	$a = b = 1$		первая ошибка
4	21	$a \leq T \leq b < 2a$		первая ошибка
5	25	нет	1 – 4	первая ошибка

Примеры

стандартный ввод	стандартный вывод
1 2 5	18 15
4 3 6	11 11
17 10 193	1596 1590

Замечание

Пояснение ко второму примеру:

1. в момент времени 0 первый бот отправляет второму сообщение A, а второй первому — B;
2. в момент времени 1 боты отправляют друг другу ответы на полученные на нулевой секунде сообщения: первый второму B(1) (ответ на B), а второй первому — A(1);
3. в момент времени 2 новых сообщений не появляется, и они отправляют друг другу ответы на полученные на первой секунде сообщения: A(2) (ответ на A(1)) и B(2);
4. в момент времени 3 будут отправлены B(3) и A(3), и одновременно с этим второй бот отправит первому новое сообщение C;
5. в момент времени 4 первый отправит второму новое сообщение D, C(1) (ответ на C) и A(4), а второй первому — B(4);
6. в момент времени 5 новых сообщений нет, боты отправляют друг другу ответы на полученные секунду назад сообщения;
7. в момент времени 6 будут отправлены ответы на сообщения с предыдущей секунды, а также второй бот отправит первому новое сообщение E.

Итого, первый бот отправил: A, B(1), A(2), B(3), D, C(1), A(4), B(5), D(2), C(3) и A(6), всего 11 сообщений.

Второй бот тоже отправил ровно 11 сообщений: B, A(1), B(2), C, A(3), B(4), D(1), C(2), A(4), E и B(6).

Задача С. Есть n стульев...

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

Влад наконец-то достиг позиции тимлида в команде, но теперь у него совсем нет времени на дорогу домой, и ему придется спать в офисе. К сожалению, не все IT-компании могут позволить себе просторный и удобный коворкинг, в котором можно подремать, поэтому Влад будет спать на офисных стульях.

В офисе есть n стульев, i -й из которых имеет высоту h_i и ширину w_i . Влад планирует выбрать любой набор офисных стульев $[i_1, i_2, \dots, i_k]$ и расположить в ряд, чтобы на них можно было лечь. Рост Влада равен H , поэтому, чтобы он мог удобно лежать, необходимо, чтобы суммарная ширина выбранных стульев была не меньше H , то есть

$$\sum_{j=1}^k w_{i_j} \geq H.$$

Очевидно, что спать на стульях разной высоты неудобно. Назовем неудобностью выбранного набора максимальную разность высот двух **соседних** стульев в ряду, то есть $\max_{j=2}^k |h_{i_j} - h_{i_{j-1}}|$. Если набор состоит из одного стула, его неудобность равна 0.

Помогите Владу выбрать набор стульев так, чтобы на ряду из них можно было лежать, а неудобность этого ряда была как можно меньше.

Формат входных данных

В первой строке ввода через пробел даны два целых числа n и H — количество стульев и рост Влада ($1 \leq n \leq 2 \cdot 10^5$; $1 \leq H \leq 10^9$).

Во второй строке ввода через пробел перечислены n целых чисел h_i — высоты стульев ($1 \leq h_i \leq 10^9$). В третьей строке в том же формате перечислены n целых чисел w_i , равных ширине стульев ($1 \leq w_i \leq 10^9$).

Гарантируется, что H не превосходит суммы всех w_i .

Формат выходных данных

Выведите единственное число — минимальное возможное неудобство среди всех подходящих наборов.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	10	$n \leq 100$		полная
2	20	$n \leq 1000$	1	первая ошибка
3	15	$w_i = 1$; $n \leq 10^5$		первая ошибка
4	19	$h_i \leq 30$; $n \leq 10^5$		первая ошибка
5	36	нет	1 – 4	первая ошибка

Примеры

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

Замечание

В первом примере нужно выставить стулья 2 и 4 в любом порядке.

Во втором примере можно выбрать, например, следующие наборы: $[1, 5]$, $[2, 4, 3]$. Обратите внимание, что порядок стульев в наборе важен: неудобность набора $[2, 3, 4]$ равна $\max(|5 - 3|, |4 - 5|) = \max(2, 1) = 2$, что больше, чем для набора $[2, 4, 3]$.

Задача D. Перекладывание ответственности

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2.5 секунд
Ограничение по памяти:	256 мегабайт

При подготовке олимпиад у каждого разработчика есть своя зона ответственности. Обычно каждый разработчик полностью отвечает за подготовку какой-то определенной задачи, однако в этот раз жюри ИОИП решило поступить по-другому.

Всего в команде разработчиков n человек. Также есть n задач, которые необходимо подготовить. Подготовка i -й задачи требует подготовки ровно c_i ее элементов, и разработка каждого элемента i -й задачи имеет сложность w_i .

Было решено, что каждый разработчик будет отвечать за столько же элементов, за сколько он бы отвечал, если бы разрабатывал целиком соответствующую задачу. Иными словами, i -му разработчику будет назначено ровно c_i элементов из различных задач. Распределение элементов по разработчикам происходит следующим образом:

1. Сначала первому разработчику выдается c_1 элементов, затем второму — c_2 , и так далее. Переход к $(i + 1)$ -му разработчику происходит в тот момент, когда i -му назначается ровно c_i элементов.
2. Элементы, за которые будет отвечать каждый разработчик, выбираются по одному из всех еще не до конца распределенных задач **по очереди**. Сначала будет выбран один этап из первой задачи, затем — из второй, из третьей, и так далее по кругу. Если в какой-то задаче не осталось нераспределенных этапов, она пропускается.
3. Элементы, назначаемые очередному разработчику, выбираются начиная с той задачи, на которой остановился предыдущий разработчик. То есть, если последний элемент, назначенный предыдущему разработчику, был из x -й задачи, то первый элемент, назначенный следующему, будет из задачи $(x + 1) \bmod n$ (если в ней еще остались нераспределенные элементы).

Иными словами, поддерживается набор еще не до конца распределенных задач и указатель x на «текущую» задачу. Когда надо выдать текущему разработчику очередной элемент, ему выдается один элемент из задачи x , после чего x сдвигается по кругу вперед на следующую задачу.

Жюри считает, что такой способ позволяет более честно распределить сложность подготовки олимпиады. Определите суммарную сложность разработки элементов, доставшихся каждому из n разработчиков.

Формат входных данных

В первой строке дано целое число n — количество разработчиков ($1 \leq n \leq 500\,000$).

В i -й из следующих n строк через пробел даны два целых числа c_i и w_i — количество элементов в i -й задаче и сложность их разработки ($1 \leq c_i, w_i \leq 10^9$).

Формат выходных данных

В единственной строке выведите через пробел n чисел, i -е из которых равно суммарной сложности разработки элементов, доставшихся i -му разработчику.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	10	$n \leq 1000; c_i \leq 1000$ для всех i		полная
2	10	$n = 2$		первая ошибка
3	13	$c_i \leq 2$ для всех i		первая ошибка
4	17	$n \leq 1000$	1, 2	первая ошибка
5	29	$n \leq 200\,000$	4	первая ошибка
6	21	нет	1 – 5	первая ошибка

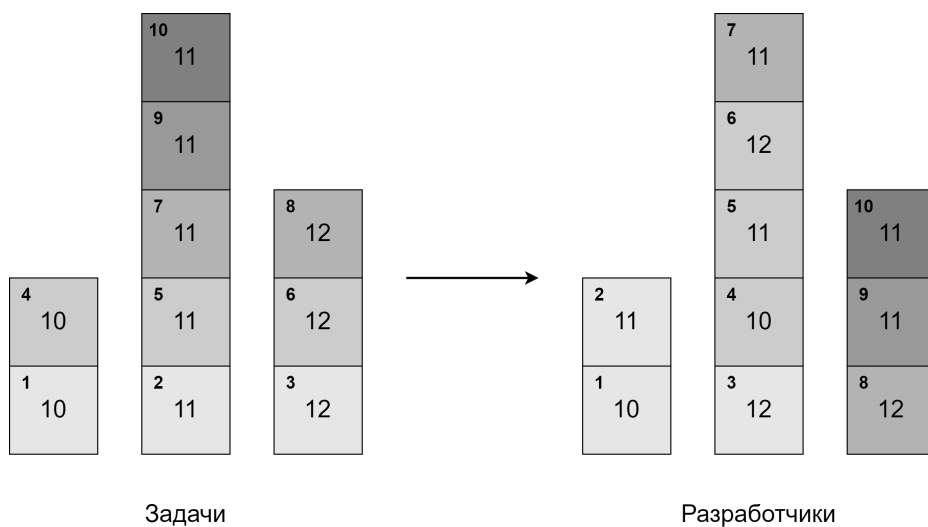
Примеры

стандартный ввод	стандартный вывод
3 3 1 2 10 4 100	111 11 301
1 10 10	100
3 2 10 5 11 3 12	21 56 34

Замечание

Иллюстрацию к третьему примеру можно видеть ниже. Слева показаны элементы, из которых состоят задачи, справа — элементы, назначенные каждому разработчику.

В центре каждого элемента указана сложность его реализации, а число в левом верхнем углу обозначает порядок выбора элементов. Элементы выбираются из задач в порядке слева-направо, затем снизу-вверх, а назначаются в порядке снизу-вверх, затем слева-направо.



Задача Е. Быстрый исполнитель

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

Студент первого курса ИТМО Миша изучает новый примитивный язык программирования. В этом языке все операции производятся над массивами целых неотрицательных чисел длины n .

Миша успел создать массив a и равный ему массив b . Также он успел реализовать четыре функции:

1. **shift** — делает циклический сдвиг массива a влево на d , то есть при $a = [a_0, a_1, \dots, a_{n-1}]$ выполняет присваивание

$$a \leftarrow [a_d, \dots, a_{n-1}, a_0, \dots, a_{d-1}];$$

2. **xor** — присваивает в массив b его поэлементный **xor** (побитовое исключающее «или») с массивом a , то есть

$$b \leftarrow [a_0 \oplus b_0, a_1 \oplus b_1, \dots, a_{n-1} \oplus b_{n-1}];$$

3. **and** — присваивает в массив b его поэлементный **and** (побитовое «и») с массивом a ;

4. **or** — присваивает в массив b его поэлементный **or** (побитовое «или») с массивом a .

Используя эти функции, Миша написал программу, задаваемую последовательностью операций **xor**, **and** и **or** длины m . Программа в цикле p раз выполняет следующие действия: для каждой операции из последовательности сначала вызывается **shift**, а затем соответствующая этой операции функция. Так, для последовательности операций **[or, xor, and]** и $p = 5$ программа будет выглядеть как

```
b = a = [...]  
repeat 5 times {  
    shift  
    or  
    shift  
    xor  
    shift  
    and  
}
```

К сожалению, язык еще новый, и его интерпретатор не справляется с выполнением такой программы. Помогите Мише определить, чему будет равно конечное состояние массива b после выполнения заданной программы.

Формат входных данных

В первой строке ввода через пробел перечислены четыре целых числа n , m , d и p — длина массива, количество операций в последовательности, величина сдвига и количество повторений ($0 \leq d < n \leq 2 \cdot 10^5$; $1 \leq m \leq 10$; $1 \leq p \leq 10^9$).

Во второй строке через пробел перечислены n целых чисел a_i — элементы массива a , они же — изначальные значения элементов массива b ($0 \leq a_i \leq 10^9$).

В третьей строке через пробел перечислены m слов, каждое из которых равно «xor», «and» или «or» — последовательность применяемых на каждой итерации цикла операций.

Формат выходных данных

Выведите в одну строку через пробел n целых чисел — элементы массива b после выполнения описанной программы.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	11	$n \leq 1000; p \leq 100$		полная
2	14	$m = 1$		первая ошибка
3	17	все m операций одинаковы	2	первая ошибка
4	15	нет операций xor		первая ошибка
5	16	$a_i \leq 1$ для всех i		первая ошибка
6	27	нет	1 – 5	первая ошибка

Примеры

стандартный ввод	стандартный вывод
5 3 2 2 1 0 1 0 1 or and or	1 0 1 1 1
6 3 2 3 1 2 3 4 5 6 xor and or	1 6 3 6 5 6
8 4 3 10 17 26 4 12 25 11 43 1 and or xor and	0 2 0 8 17 1 9 1
10 4 8 10 9 4 4 5 13 2 2 11 0 12 or xor xor xor	2 8 9 0 6 1 13 6 0 11

Задача F. Устный счет

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	1024 мегабайта

Отвлечемся от задач про программистов и перенесемся в совершенно обыкновенные ясли, где мальчик Вова (3 годика) практикуется в устном счете, а если точнее — в вычислении арифметических выражений по модулю $10^9 + 7$.

Совсем недавно Вова придумал длинное и очень красивое арифметическое выражение. Выражение состояло из n целых неотрицательных чисел, меньших $10^9 + 7$, и знаков сложения и умножения между ними. Первым же делом он вычислил это выражение по модулю $10^9 + 7$, после чего выписал само выражение и результат на лист бумаги. Все числа он выписал без ведущих нулей.

Пока шел тихий час, Вова спал, а хулиганы стерли несколько цифр в выражении и заменили их на другие. Менять знаки или результат вычисления выражения хулиганы не решились, даже для них это слишком низко. Когда Вова проснулся, он обнаружил рядом с его листочком ластик и карандаш. У Вовы есть принципы — если хулиганы изменили не более двух цифр, то он их простит, иначе они будут наказаны по всей строгости ясельных законов.

К сожалению, чтобы понять, сколько цифр изменили хулиганы, Вове явно понадобится ваша помощь. Определите, могло ли данное выражение быть получено из верного равенства заменой не более двух цифр, и если да, то в каких числах и какие цифры были изменены.

Формат входных данных

Единственная строка входного файла содержит выражение, которое обнаружил Вова у себя на листке. Выражение состоит из двух частей, разделенных знаком '='.

Первая часть содержит n целых неотрицательных чисел, разделенных знаками сложения ('+') и умножения ('*') ($1 \leq n \leq 10^5$). Вторая часть — целое неотрицательное число, означающее результат вычисления выражения.

Числа и все знаки разделяются одним пробелом. Гарантируется, что все числа строго меньше $10^9 + 7$ и записаны без ведущих нулей.

Формат выходных данных

В случае, если данное выражение не может быть получено из верного равенства заменой не более двух цифр, выведите «NO» (без кавычек).

В противном случае в первой строке выведите «YES». На следующей строке выведите целое число $k \leq 2$ — количество **чисел**, которые были изменены.

В следующих k строках выведите по два числа — позицию измененного числа в выражении (от 1 до n) и его исходное значение.

Суммарно должно быть изменено не более двух цифр. В процессе замены в числах не должны появиться ведущие нули и числа должны остаться меньше $10^9 + 7$. Если существует несколько вариантов ответа, выведите любой из них.

Система оценки

Баллы за подзадачи 1 – 6 начисляются только в случае, если все тесты соответствующей подзадачи и необходимых подзадач, а также тесты из условия, успешно пройдены.

Подзадача 7 оценивается потестово — всего в ней 25 тестов, каждый из которых независимо оценивается в 1 балл.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	7	$n \leq 45$		первая ошибка
2	13	$n \leq 100$	1	первая ошибка
3	15	все числа в выражении меньше 10		первая ошибка
4	12	нет операций умножения		первая ошибка
5	13	нет операций сложения		первая ошибка
6	15	все числа в выражении меньше 10^5	3	первая ошибка
7	25	нет	1 – 6	потестовая оценка

В подзадачах 3 и 6 ограничение касается только чисел слева от знака равенства, результат вычисления выражения может быть произвольным числом, меньшим $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
$56 + 14 * 86 + 51 * 55 = 3925$	YES 1 3 76
$97 + 14 * 31 * 76 + 99 * 73 = 40930$	NO