

Задача А. Petya++

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

Около года назад Petya++ сразился с Гаменом — моноблоком с искусственным интеллектом, который продвигал свою компанию за счет кражи чужих технологий и устранения их авторов. Гамен схватил всех его друзей, и поэтому Petya++ был вынужден пойти на сделку: он остался с Гаменом, и тот использовал его мощности для сложнейших расчетов. Взамен Гамен обещал отпустить его друзей и оставить их в покое.

Комната, где содержали Petya++, была снаружи закрыта палиндромным замком. Такой замок открывается, если введенная в него строка является палиндромом, т. е. читается одинаково как слева направо, так и справа налево. Petya++ понимал, что Гамен не сдержит свое обещание и попытается устранить людей, которые знают правду о темной стороне его компании. Поэтому через некоторое время он решил совершить побег.



Petya++ каким-то образом смог узнать текущую комбинацию, введенную на замке. Это была строка s длины n , состоящая из единиц и нулей. Для вскрытия замка он использовал свою новую роборуку: запуская специальные импульсы в стену, он мог модифицировать эту строку, используя операции одного из двух типов:

1. Вставить строку 110 в строку в качестве подпоследовательности. Это значит, что в исходную строку на некоторую позицию (возможно, перед первым элементом или после последнего) добавляется единица. Далее добавляется еще одна единица, но уже только на позицию, находящуюся где-то после первой добавленной единицы в строке. После этого, по подобному принципу, добавляется ноль на позицию правее последней единицы. Необязательно, чтобы в итоге добавленные символы шли подряд, однако они обязаны сохранять относительный порядок.
2. Вставить $0x1$ в качестве подпоследовательности, где x — некоторый заданный символ (ноль или единица).

Напоминаем, что для открытия замка необходимо сделать так, чтобы введенная в него комбинация стала палиндромом.

Petya++ успешно открыл замок за не более чем n действий. После удачного изменения строки он незаметно выбрался из комплекса, замаскировался и начал собирать улики для разоблачения Гамена.

А можете ли вы предположить, какие именно сигналы отправлял Petya++, чтобы замок открылся? Если существует несколько ответов, предоставьте любой из них.

Формат входных данных

Первая строка входных данных содержит одно целое число n ($1 \leq n \leq 10^3$) — длина строки.

Вторая строка входных данных содержит строку s , состоящую из символов из символов «0» и «1» — введенная в замок комбинация.

Третья строка входных данных содержит символ x («0» или «1»), обозначающий, что разрешено вставлять строку $0x1$ в качестве подпоследовательности.

Формат выходных данных

В первой строке выведите одно целое число m ($0 \leq m \leq n$) — количество операций, которые необходимо сделать, чтобы последовательность на замке стала палиндромом.

В каждой из следующих m строк выведите описание очередной операции в виде четырех целых чисел t, pos_1, pos_2 и pos_3 ($1 \leq t \leq 2, 0 \leq pos_1 \leq pos_2 \leq pos_3 \leq |s|$), где t — тип операции, а pos_1, pos_2, pos_3 — после какого символа в строке мы вставляем очередной символ подпоследовательности. Символы в строке нумеруются, начиная с единицы, при этом, если вставка i -го символа происходит в начало строки, то pos_i должен быть равен нулю. Иными словами, pos_i обозначает, сколько символов строки, которая была до выполнения текущей операции, находится до i -го символа вставленной подпоследовательности. Например, если строка равна `101`, и мы вставляем `011` с $pos_1 = 1, pos_2 = 1$ и $pos_3 = 3$, то получится `101011`.

Система оценки

В подзадачах, где задано ограничение на M , дополнительно гарантируется, что задачу можно решить, сделав не более чем M операций.

№	Дополнительные ограничения	Баллы за подзадачу	Необходимые подзадачи
1	Строка состоит из нулей	14	
2	$n \leq 10, M = 1$	21	
3	$x = 1, n \leq 3$	8	
4	$x = 1$	19	3
5	$x = 0, n \leq 3$	9	
6	$x = 0$	29	5

Примеры

стандартный ввод	стандартный вывод
6 100110 0	1 2 4 6 6
3 110 1	1 2 1 2 3

Замечание

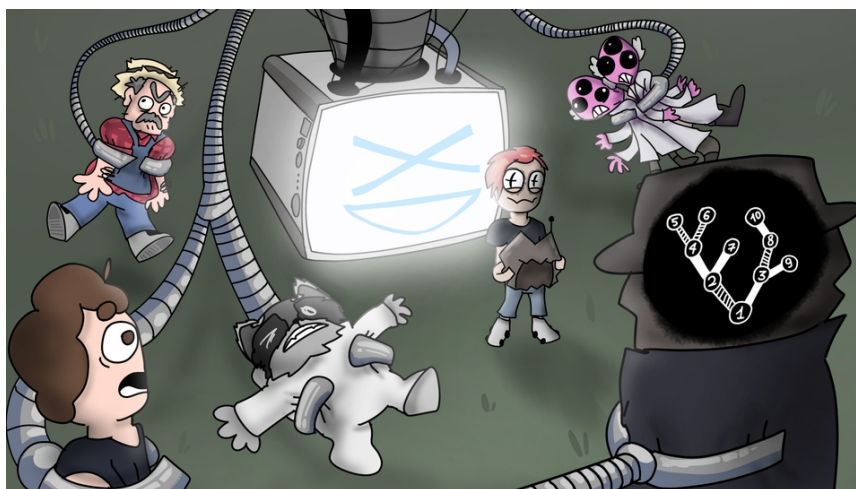
В первом тестовом примере после выполнения всех операций получается строка `100101001`.

Во втором тестовом примере после выполнения всех операций получается строка `101101`.

Задача В. Безвыходная ситуация

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

Но как же Петя мог упустить, что Petya++ не вернулся домой? Чтобы не вызывать подозрений, Гамен отправил вместе с ним органического двойника Petya++ — Лжепетю, который незаметно для всех заменял робота последний год. Именно Лжепетя отправил Казимиру коробку с паразитами и пытался заманить Петю в ловушку, пока Petya++ пытался остановить двойника. К сожалению, помешать Лжепете не удалось, и в суматохе Гамену, который скрывался на месте заготовленной ловушки, удалось схватить наших героев. Ситуация казалась безвыходной, и Petya++ погрузился в чертоги разума в поисках решения.



Чертоги разума робота представляют собой n областей. Некоторые пары областей могут быть связаны между собой коридором, всего коридоров $n-1$. Чертоги представляют собой связную структуру. Это означает, что, начиная из любой области, проходя через некоторое количество коридоров, всегда можно добраться до любой другой области.

Изначально чертоги деактивированы, и их коридоры неактивны. Для настройки на некоторую задачу каждый коридор должен быть активирован в определенном *ритме*: положительном или отрицательном. Активация коридоров происходит следующим образом. Petya++ выбирает, из какой области чертогов он хочет начать активацию. Далее, за один шаг он должен перейти из одной области в другую по некоторому коридору. Если по коридору перешли на четном шаге, то он активируется в положительном ритме, а если на нечетном — в отрицательном. Это значит, что если, например, на первом шаге мы перешли через некоторый коридор, то он активируется в отрицательном ритме, а если на четвертом шаге — то в положительном.

Petya++ может любое количество раз проходить по одному и тому же коридору. В таком случае он будет активирован в ритме последнего шага, на котором через него прошли.

Для настройки на поиск выхода из данной ситуации Petya++ должен для каждого коридора с номером i активировать его в ритме a_i . Помогите роботу узнать, может ли он настроиться на решение этой задачи со своим устройством чертогов разума.

Формат входных данных

В первой строке входных данных находится целое число t ($1 \leq t \leq 1000$) — количество тестовых примеров.

Далее следует t тестовых примеров, заданных описанным ниже способом.

В первой строке тестового примера находится два целых числа n ($1 \leq n \leq 5 \cdot 10^5$) — количество областей.

В каждой из следующих $n-1$ строк тестового примера находится по два целых числа u_i и v_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$) — коридоры, соединяющие чертоги.

В следующей строке тестового примера содержится $n - 1$ целых чисел a_i ($0 \leq a_i \leq 1$) — ритм, в котором должен быть активирован i -й коридор. Здесь 0 означает положительный ритм, а 1 — отрицательный.

Обозначим сумму n по всем тестовым примерам как S_n . Тогда гарантируется, что $S_n \leq 5 \cdot 10^5$.

Формат выходных данных

Для каждого из тестовых примеров выводите ответ в отдельной строке. Каждая строка должна содержать одно слово — «Yes» (без кавычек), если возможно активировать коридоры в необходимом ритме, или «No» (без кавычек) в противном случае.

Система оценки

№	Дополнительные ограничения	Баллы за подзадачу	Необходимые подзадачи
1	$S_n \leq 500, u_i = i, v_i = i + 1$	6	
2	$S_n \leq 2 \cdot 10^3, u_i = i, v_i = i + 1$	9	1
3	$u_i = i, v_i = i + 1$	16	1-2
4	$S_n \leq 500$	9	1
5	$S_n \leq 2 \cdot 10^3$	22	1-2, 4
6	$a_i = 0$ для любого i	7	
7	$a_i = 1$ для любого i	5	
8	Нет дополнительных ограничений	26	1-7

Пример

стандартный ввод	стандартный вывод
2	Yes
10	No
1 2	
1 3	
2 4	
2 7	
4 5	
4 6	
3 8	
3 9	
8 10	
0 1 1 1 0 0 0 1 1	
4	
1 2	
2 3	
3 4	
1 1 1	

Замечание

В первом тестовом примере все коридоры будут активированы в необходимом ритме, если мы пройдем по чертогам в следующем порядке: 1 — 2 — 4 — 5 — 4 — 6 — 4 — 2 — 7 — 2 — 1 — 3 — 9 — 3 — 8 — 10. Отметим, что это не единственное решение, например, подойдет такой же обход, но в обратном порядке.

Во втором тестовом примере можно показать, что не существует способа обойти чертоги, чтобы все коридоры были активированы в отрицательном ритме.

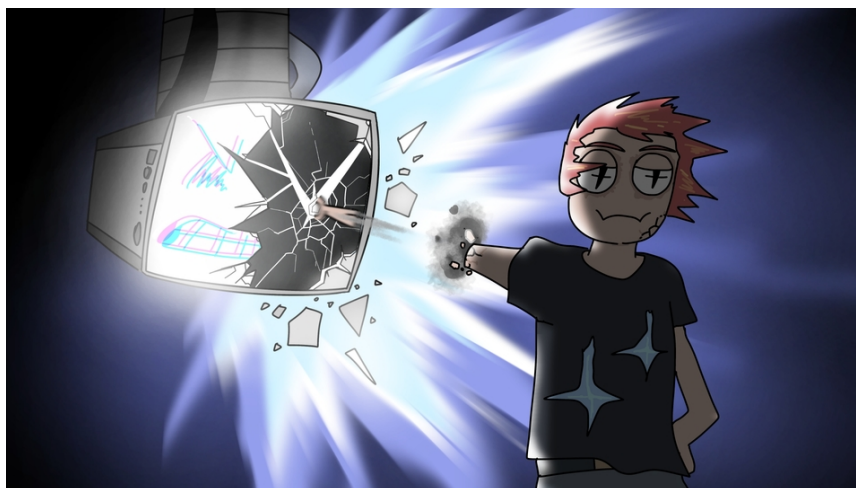
Задача С. Выход

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Petya++ не смог придумать никакого решения: в этот раз он не сможет всех спасти. Гамен уже зарядил свою пушку, и казалось, что это конец. . .

. . . Но в последнюю секунду Лжепетя выстрелил своей фальшивой роборукой прямо в моноблок, устроив тому короткое замыкание!

Зачем он это сделал и чего он хочет? Petya++ решил, что еще будет время об этом подумать. Сначала нужно освободиться и освободить друзей.



Petya++ схвачен манипулятором Гамена, который, несмотря на замыкание, продолжает работать и крепко его держать. Манипулятор состоит из $n + 1$ конденсаторов, пронумерованных от 0 до n . При этом $(i - 1)$ -й и i -й конденсаторы ($1 \leq i \leq n$) соединены друг с другом, и между ними стоит транзистор с номером i . Таким образом, в цепи всего n транзисторов, пронумерованных от 1 до n . Благодаря короткому замыканию на t конденсаторах образовалась *избыточная энергия*, при этом на конденсаторе с номером p_i избыточная энергия равна a_i единиц. Гарантируется, что все p_i различны.

Для каждого из конденсаторов Petya++ может перенаправить избыточную энергию либо влево по цепи, либо вправо. Если перенаправить ее с конденсатора номер p_i , то перенаправленная энергия моментально устроит перегорание a_i последовательных соседних транзисторов в заданном направлении, начиная с текущего. Если в заданном направлении нет a_i транзисторов, то энергия устроит перегорание во всех оставшихся транзисторах в данном направлении. К сожалению, он не может отправить какую-то часть энергии в одну сторону, а какую-то в другую: после изначального выбора направления вся энергия уходит в него. Затем перенаправленная энергия исчезает, поэтому использовать один конденсатор дважды также не выйдет. Дополнительно отметим, что если транзистор перегорел, то он все еще участвует в цепи и никак не препятствует прохождению энергии.

Например, если на конденсаторе номер 5 накопилась избыточная энергия 3, то при перепаравлении влево перегорят транзисторы номер 5, 4 и 3, а при перенаправлении вправо — номер 6, 7 и 8. Если же, например, транзистор номер 7 перегорел еще до применения этой операции, а мы перенаправили энергию вправо, то перегорят транзисторы номер 6 и 8 (поскольку транзистор номер 7 перегорел еще раньше), а, например, транзистор номер 9 затронут не будет.

Petya++ подозревает, что чем больше транзисторов перегорит, тем меньше шанс, что манипулятор будет продолжать работать. Поэтому робот задался вопросом: в какую сторону лучше перенаправить энергию из каждого конденсатора, чтобы суммарное количество перегоревших транзисторов было как можно больше?

Формат входных данных

В первой строке входных данных содержится два целых числа n и m ($1 \leq n \leq 10^6$, $1 \leq m \leq 500$) — наибольший номер конденсатора и количество конденсаторов, на которых изначально образовалась избыточная энергия.

В следующих m строках входных данных содержатся два целых числа p_i и a_i ($0 \leq p_i \leq n$, $a_i \leq n$) — позиция конденсатора, на котором изначально образовалась избыточная энергия и количество этой избыточной энергии.

Гарантируется, что все p_i различны.

Формат выходных данных

Выведите одно целое число — максимальное количество транзисторов, которые могут перегореть в результате действий Petya++.

Система оценки

№	Дополнительные ограничения	Баллы за подзадачу	Необходимые подзадачи
1	$m \leq 20$	7	
2	$m \leq 30$	9	1
3	$m \leq 35$	8	1 - 2
4	$m \leq 40$	7	1 - 3
5	$m \leq 50$	6	1 - 4
6	$m \leq 100$	10	1 - 5
7	$m \leq 200$	7	1 - 6
8	$a_i = 1$	5	
9	$a_i \leq 2$	11	8
10	$n \leq 2500$, $a_i \leq 7$	18	
11	Нет дополнительных ограничений	12	1 - 10

Пример

стандартный ввод	стандартный вывод
12 3 3 6 5 5 12 2	10

Замечание

В примере из условия требуется совершить следующие действия:

- Перенаправить энергию из конденсатора на позиции 3 влево. В результате сгорят транзисторы номер 1, 2 и 3.
- Перенаправить энергию из конденсатора на позиции 5 вправо. В результате сгорят транзисторы номер 6, 7, 8, 9 и 10.
- Перенаправить энергию из конденсатора на позиции 12 влево. В результате сгорят транзисторы номер 11 и 12.

Итого выходит, что сгорело 10 транзисторов. Можно показать, что данный ответ оптимален.

Задача D. Договор на будущее

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Лжепетя помог Petya++ и его друзьям выбраться из лап манипулятора. Он объяснил, что, увидев все злодеяния Гамена, не мог оставаться на его стороне. Теперь, когда моноблок повержен, *Turbopallascats* юридически перейдет под управление Лжепетя. Тот пообещал, что в будущем компания больше никому не доставит проблем: никаких смертельных выставок и краж технологий. На этом они и договорились, и столь долгое и захватывающее приключение, наконец-то, по-настоящему подошло к концу.



Однако у Казимира все еще оставались вопросы к Petya++: хоть и с благими намерениями, но именно он уничтожил киберцпак. А также зашифровал письма, сломал телефон, отправил на свалку посылку (пусть и зараженную паразитами) и сделал много чего еще. За роботом явно остался должок. Тот заверил, что обязательно все возместит, и, ни в коем случае не меняя тему, предложил друзьям сначала решить загадку.

Petya++ называет длинное число, состоящее из n цифр. Но цифры эти необычные: они заданы не в десятичной системе, а в 2^{20} -ичной! Это означает, что каждая цифра представлена целым числом от 0 до $2^{20} - 1$.

Робот держит это число в голове и производит с ним некоторые действия:

1. Прибавить к *каждой цифре* данного числа некоторое число x_i ($0 \leq x_i \leq 2^{20} - 1$). Если какая-то цифра становится больше или равной 2^{20} , то от нее отнимается 2^{20} .
2. Сделать циклический сдвиг цифр этого числа влево на k_i позиций, т. е. взять отрезок из первых k_i цифр, удалить его из начала и вставить в конец числа.
3. Спросить у друзей, каково значение полученного после предыдущих действий числа, то есть сумму $2^{20p} \cdot a_p$ по всем $0 \leq p \leq n - 1$. Так как это значение может быть слишком большим, Petya++ хочет узнать его остаток от деления на некоторое небольшое число d_i .

Petya++ обещал, что если на все вопросы друзья ответят верно, то он обязательно вернет долг. А вот чтобы это проверить, друзьям уже понадобится ваша помощь.

Формат входных данных

В первой строке входных данных находится два целых числа n и q ($1 \leq n, q \leq 5 \cdot 10^5$) — количество цифр в числе и количество действий соответственно.

Во второй строке входных данных находится n целых чисел a_i ($0 \leq a_i \leq 2^{20} - 1$) — цифры числа. Нумерация цифр начинается с нуля.

В каждой из следующих q строк входных данных дано описание действий Petya++. Действия бывают одного из трех типов:

- $+$ x_i ($0 \leq x_i \leq 2^{20} - 1$) — прибавить ко всем цифрам x_i ;
- $<$ k_i ($1 \leq k_i \leq n - 1$) — циклически сдвинуть число влево на k_i позиций;
- $?$ d_i ($2 \leq d_i \leq 20$) — узнать остаток от деления значения числа на d_i .

Формат выходных данных

Для каждого действия третьего типа выведите одно число в отдельной строке — искомый остаток от деления.

Система оценки

№	Дополнительные ограничения	Баллы за подзадачу	Необходимые подзадачи
1	$n, q \leq 1000$	10	
2	$d_i \in \{2, 4, 8, 16\}$	4	
3	$d_i = 11$	8	
4	Все действия типа $<$ и $?$	14	
5	Все действия типа $+$ и $?$	26	
6	$a_i \leq 2^{19} - 1, \sum x_i \leq 2^{19} - 1$	8	
7	$n, q \leq 2 \cdot 10^5$	18	1
8	Нет дополнительных ограничений	12	1 - 7

Пример

стандартный ввод	стандартный вывод
4 10	8
1 5 4 8	13
? 19	0
? 20	8
+ 3	1
? 2	2
? 11	11
< 3	
? 14	
+ 1048575	
? 12	
? 15	

Замечание

«В будущем-то да, в будущем... Да...» — тихо, себе под нос, говорил Лжепетя.

